DEVELOPMENT OF
A COMPUTERISED PRODUCTION PLANNING AND CONTROL SYSTEM
FOR A SMALL MANUFACTURING COMPANY

A thesis

submitted in fulfilment

of the requirements for the Degree

of

Master of Engineering

in the

University of Canterbury

by

W. Y. Chee

University of Canterbury

1990

To my mother and father

ABSTRACT

In this project a personal computer-based Oracle software package was used as a development tool for a material requirements planning (MRP) system for a small company. The main objective was to find the right approach and techniques to use in developing such a system.

The system and structured design approaches were followed. The emphasis was on the use of graphic tools for designing. Three diagramming tools were used at the system design stage: decomposition, entity-relationship and data flow diagrams.

A relational database approach was used for data organisation. It was necessary to use the normalisation process to design the tables in the database.

Test runs of the designed system showed that the MRP system was working and with Oracle as the software platform it was flexible to changes.

ACKNOWLEDGEMENTS

I wish to thank my supervisor, Dr. Graeme Britton, for getting the project started and for his guidance, encouragement and help. He introduced and taught me the right approaches to system development. His dynamic approach to his work, is an inspiration to me.

I would like to thank Professor McCallion for editing my final draft.

To my friends: I owe much to Dr. T. L. Tai for spending many nights in editing and proof-reading all the chapters. I sincerely thank Soh Tan and Boon Wah for their much needed support. Much thanks to Quak Yoong for his help in getting the thesis printed. And to the many other friends who have helped me, I thank you.

Last, but most important, I thank my partner, Jit Lin, for her patience, encouragement and support.

TABLE OF CONTENTS

CHAPTER			PAGE
	ABSTRACT		III
	ACKNOWLEDGEMENTS		IV
	LIST OF FIGURES		IX
I	INTRODUCTION		
	1.1	Project's Origin	1
	1.2	Stages in the System Development	3
	1.3	Structured Design	4
	1.4	The System Approach	6
II	ANALYSIS OF THE BUSINESS AND ITS PROBLEMS		
	2.1	A General Description of the Business	8
	2.2	Organisation Structure	9
	2.3	Factory Layout and Material Flow	11
	2.4	Information Flow	13
	2.5	Deficiencies in the Organisation	17

III ESSENTIAL FEATURES OF AN MRP SYSTEM

3.1 Types of MRP System 21

3.2 Essential Features 23

3.2.1 Master production schedule 23

3.2.2 Bills of materials 24

3.2.3 Capacity planning 25

3.2.4 Computation of material Requirement 26

IV MODELLING THE SYSTEM

4.1 Decomposition Diagrams 30

4.2 Entity-relationship Diagrams 34

4.3 Data Flow Diagrams 36

V INTRODUCTION TO RELATIONAL DATABASE, 4GL AND ORACLE SOFTWARES

5.1 Relational Database 39

5.2 Fourth Generation Language (4GL) 43

5.3 Oracle Products 47

VI BUILDING THE SYSTEM

6.1 Building the Database 51

6.1.1 Primary and foreign keys 51

6.1.2 Flat table 52

6.1.3 Duplicated data 53

6.1.4 Normalisation processes 54

6.1.5 SQL*plus 57

6.2 Building the User Interface 56

6.2.1 Building the forms 56

6.2.2 Building the menu 76

VII EVALUATION

7.1 System Capability 78

7.2 Approaches to the System Development 78

7.3 Diagramming Techniques 79

7.4 Oracle Software Package 80

7.4.1 Oracle Flexibility 80

7.4.2 Ease of Use and Learning 81

7.4.3 Processing Speed 82

7.5 End User Development 82

7.6 Conclusion 83

REFERENCES 86

BIBLIOGRAPHY 87

APPENDICES

A.	Entity-relationship Diagram Keys	88
B.	Detail Data Flow Diagrams	90
C.	Ship Schedules	96
D.	4GL Report Generators Comparison	97
E.	Oracle Hardware Requirements	98
F.	SQL*Forms Terminology	99
G.	First, Second and Third Normal Forms	100
H.	Tables in the Database	103
I.	SQL*Menu Documentation	106
J.	Detail Description of Forms	114
K	SQL*Forms Triggers	147

LIST OF FIGURES

FIGURE		PAGE
1	Stages in the system development	3
2	Organisation structure	9
3	Factory plan	12
4	Information and material flow diagram	14
5	Closed-loop MRP system	22
6	Product structure tree example	25
7	Decomposition diagram	32
8	Entity-relationship diagram	35
9	Data flow diagram	37
10	Application-centred data organisation	41
11	Database approach	41
12	Relational table attributes	42
13	Application size to software technology match	44
14	Computer software generations	45
15	Oracle software products	48
16	Page 1 of Master form	60
17	Page 2 of Master form	60
18	Page 3 of Master form	61
19	Page 4 of Master form	61
20	Page 1 of Creorder form	62
21	Page 2 of Creorder form	63
22	Page 3 of Creorder form	63

23	Page 1 of Capacity form	65
24	Page 2 of Capacity form	65
25	Page 3 of Capacity form	66
26	Page 4 of Capacity form	66
27	Page 5 of Capacity form	67
28	Page 6 of Capacity form	67
29	Page 1 of Factory form	68
30	Page 2 of Factory form	69
31	Page 3 of Factory form	69
32	Page 4 of Factory form	70
33	Page 5 of Factory form	70
34	Page 1 of Reorder form	71
35	Page 2 of Reorder form	72
36	Page 1 of Control form	73
37	Page 2 of Control form	73
38	Page 1 of Performance form	74
39	Page 2 of Performance form	74
40	SQL*menu-application specification	76
41	SQL*menu-work class/user information	76
42	SQL*menu-options for application	77
B1	DFD of shift progress data entry	90
B2	DFD of processing order	91
B3	DFD of committing order	92
B4	DFD of scheduling	93
B5	DFD of reports	94
B6	DFD of calculating performance data	94
B7	DFD of inwards/outwards goods data entry	95

B8	DFD of processing purchase form	95
J1	Page 1 of Master form	122
J2	Page 3 of Master form	123
J3	Page 1 of Creorder form	123
J4	Page 5 of Capacity form	129
J5	Page 3 of Master form	131
J6	Page 4 of Capacity form	131
J7	Page 1 of Factory form	132
J8	Query statement for stock level	139
J9	Query statement for WIP	139
J10	Material reorder proposals	141

CHAPTER I

INTRODUCTION

1.1 Project's Origin

This project concerned the development of a management system for a manufacturing company in Christchurch that makes quality plastic wares. An analysis of the company was made and a proposal was to be put forward for a PC based computer system that would give the management information for production planning and control and also for processing orders from customers. At an early stage it was envisaged that the system would be built on a spreadsheet or a database management system (DBMS). The planning and control would be done through a material requirements planning (MRP) closed loop system. This would have given the management the functions that they wanted in the system at a cost lower than commercially available manufacturing software systems.

Unknown to the project developers, the company was in deep financial trouble. Mid way through the project the company went into receivership. Funds were cut from the project. It was decided that the project should continue but with a new focus. This is explained below.

The needs of this company are similar to many small manufacturers in New Zealand. Many realise the importance of information systems in improving manufacturing productivity. In a survey of New Zealand manufacturers, the most frequently cited critical issues in information systems are lower product costs, improved material and production control and improved financial control [1]. These needs may be met with a computerised MRP system.

In 1988 there were approximately 16,000 manufacturing sites in New Zealand. Only about 300 of them used an integrated manufacturing software package [1]. Small manufacturers may not need the complicated manufacturing packages currently available in the market. Also they may not be able to afford the high price of these packages. Enquiries in Christchurch revealed that a typical cost for manufacturing and inventory modules was $ 10,000 or more. Many small manufacturers may want an alternative.

The alternative may be provided by information systems developed from common development softwares such as Oracle. These softwares are getting increasingly cheaper and more user-friendly. The new focus of this project was to investigate the approach and techniques in designing and building such information system, in particular an MRP system. This could then provide a guide-line to manufacturers who may want to build such a system.

1.2 Stages in the System Development

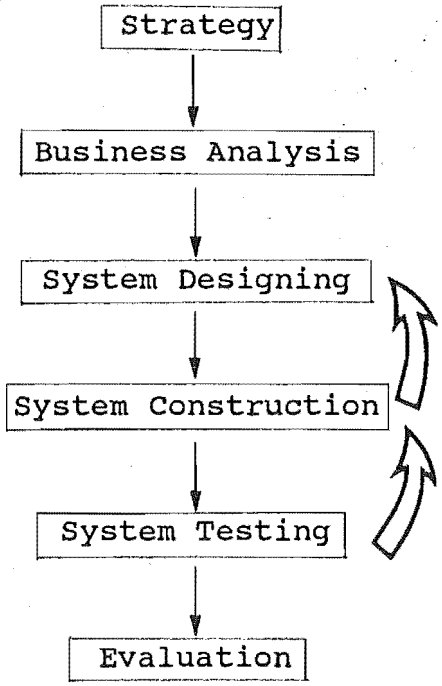


Figure 1 Stages in the system development

The different stages in the system development are shown in figure 1. The philosophical approach to the system development was investigated at the strategy stage (subsection 1.3 and 1.4).

The next stage involved a detail analysis of the business and its problems (chapter 2). It revealed several deficiencies in the management and information areas.

At the system designing stage, possible solutions to the deficiencies were investigated. Firstly, a study was carried out on the application of the MRP system for production planning and control (chapter 3). Next, three diagramming techniques were employed to model the system. The diagrams provided a graphic illustration of the necessary components and data requirements in the system (chapter 4). An assessment was then made on the type of software packages that would suit the system (chapter 5).

The construction of the system involved building a database and end-user applications (chapter 6). The system testing provided feed-back on the necessary changes and improvements to be made at the system designing and construction stages.

The final stage was an evaluation of the system development (chapter 7).

1.3 Structured Design

The approach taken in developing this system has been along the philosophies and tools used in structured design and also in the system approach. The benefits from using such techniques are explained below.

Structured design is defined as the development of a blueprint of a computer system solution to a problem, having the same components and interrelationships among the components as the original problem has [2]. To do this successfully the designer must resists making decisions on how the problem is to be solved until what the problem is has been determined.

Structured design involves a top-down approach where large, complex problems are broken down into smaller, less complex components. Decomposition continues until the original problem has been expressed as some combination of many small solvable components. The strategy is also to code and test the higher level modules first before moving on to design more detailed lower level modules. To do this dummy routines are substituted for detailed modules. These routines usually consist of returning constant output, returning a random number, immediate exit or output message to indicate that the module has been executed.

The benefits of the top-down approach include:

i) Skeleton version of the system can be developed and demonstrated to users at an early stage. This is important as users often may not know exactly what they want until they see something more tangible. Users can give valuable feedback as to the changes that they

want. Changes at this early stage are certainly much easier to make than when detailed codes have been completed.

ii) To the system developer, seeing the resulting progress at an earlier stage can be an important morale booster, especially for projects requiring a long time to complete.

iii) Debugging is easier and more systematic. This is because top-down testing tends to be incremental in nature. It consists of adding one new module to an existing skeleton of modules and observing the behaviour of the new system. If anything is wrong the problem is usually located in the new module or in the interface.

Structured design emphasises the use of graphic tools for analysis and design. It recognises that graphic tools are a very good aid in explaining systems that are technical in nature. Users find it easier to understand and to recommend any changes to a system specification when it is presented graphically.

1.4 The System Approach

In simple terms, the system approach is a philosophy which emphasises the role of a component in a larger system of components. The effectiveness of components considered collectively as a system may be greater than the sum of the effectiveness of each component considered separately.

The basic objective of the system approach to information system design is to optimise the organisation as a whole. This means that the system analyst must not look at problems in isolation from the larger system they are in. For example, proposing a just-in-time system to solve production planning and control problems would be disastrous without also solving problems dealing with unreliable suppliers.

The system approach should be taken in tandem with the structured design approach. They are complimentary to each other.

CHAPTER II

ANALYSIS OF THE BUSINESS AND ITS PROBLEMS

The system analysis started with a series of interviews with the management and also with the workers. Considerable time was spent observing the day to day running of the business and sometimes helping to do some of the functions. The objective was to have a total view of the organisation. This included understanding its constraints, the interrelationship between departments, the flow of information and the particular problems that the management wanted the system to solve.

The findings are summarised and documented below.

2.1 A General Description of the Business

The business operated as a number of companies. For instance, the factory that was involved in manufacturing the products and distribution to markets was set up as one company, and there was another which took care of overseas markets. The products manufactured included a wide range of plastic kitchen wares, food and drink serving wares and airtight storage boxes and canisters.

All the plastic parts of these products were made on fourteen injection moulding machines. Other metal parts,

raw materials and packaging materials were purchased from suppliers.

2.2 Organisation Structure

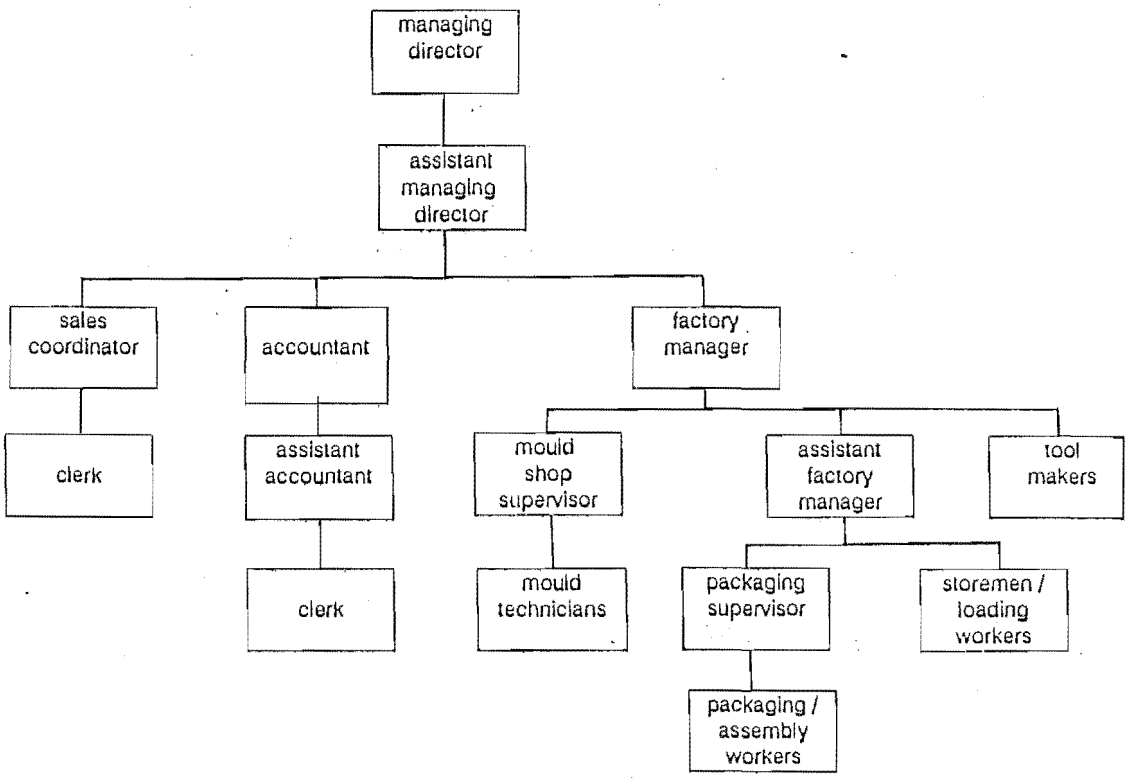


Figure 2. Organisation structure

As the number of employees was small the areas of responsibility among them were rather broad and overlapping in certain areas.

The managing director was more involved in overseas marketing. He made the initial contact with clients and often went overseas to do that. The assistant managing director would oversee the day to day running of the whole business. She also played the role of marketing, making decisions on the prices and even purchasing some of the materials.

The sales co-ordinator was in charge of processing orders, responding to local customer enquiries and dealing with the sales representatives.

The financial and accounting part of the business was the responsibilities of the accountant and his assistants.

The factory manager was responsible for the manufacturing part of the business, making sure the factory capacity could cope with the local and overseas orders. His assistant would take care of the day to day scheduling of the machines, packaging and loading processes. Supervision of all die changes and preparation of the work priority sheet (for die changes) were the mould shop supervisor's responsibilities The three of them had been working closely together in planning and controlling the whole factory production. They also purchased the materials. Effectively, the three of them were the planners for the factory.

The factory operated for three shifts a day. Each shift had its packaging supervisor who co-ordinated the assembly and packaging work. The two storemen did the store keeping and loading of products onto transport vehicles.

2.3 Factory Layout and Material Flow

The business operated from a two-story building, with the administration occupying the top level and the factory the ground level. The plan for the factory is shown in figure 3.

The movement of material is as followed (see both figures 3 and 4) :

In-coming plastic raw materials in small quantity were stored in the factory itself; larger quantity were stored in the storage building. Other materials were stored in the loading or stock areas.

The plastic materials were fork-lifted or manually brought to the mixing area to be mixed with colourant pallets for the particular colours of the products. This was done in rotating drums. The right mixtures were then manually loaded onto the moulding machine hoppers. Moulded parts were ejected out from their moulds to be picked up by robotic arms with suction pads and released onto the conveyor belt, as in machines 1, 2, 12 and 13,

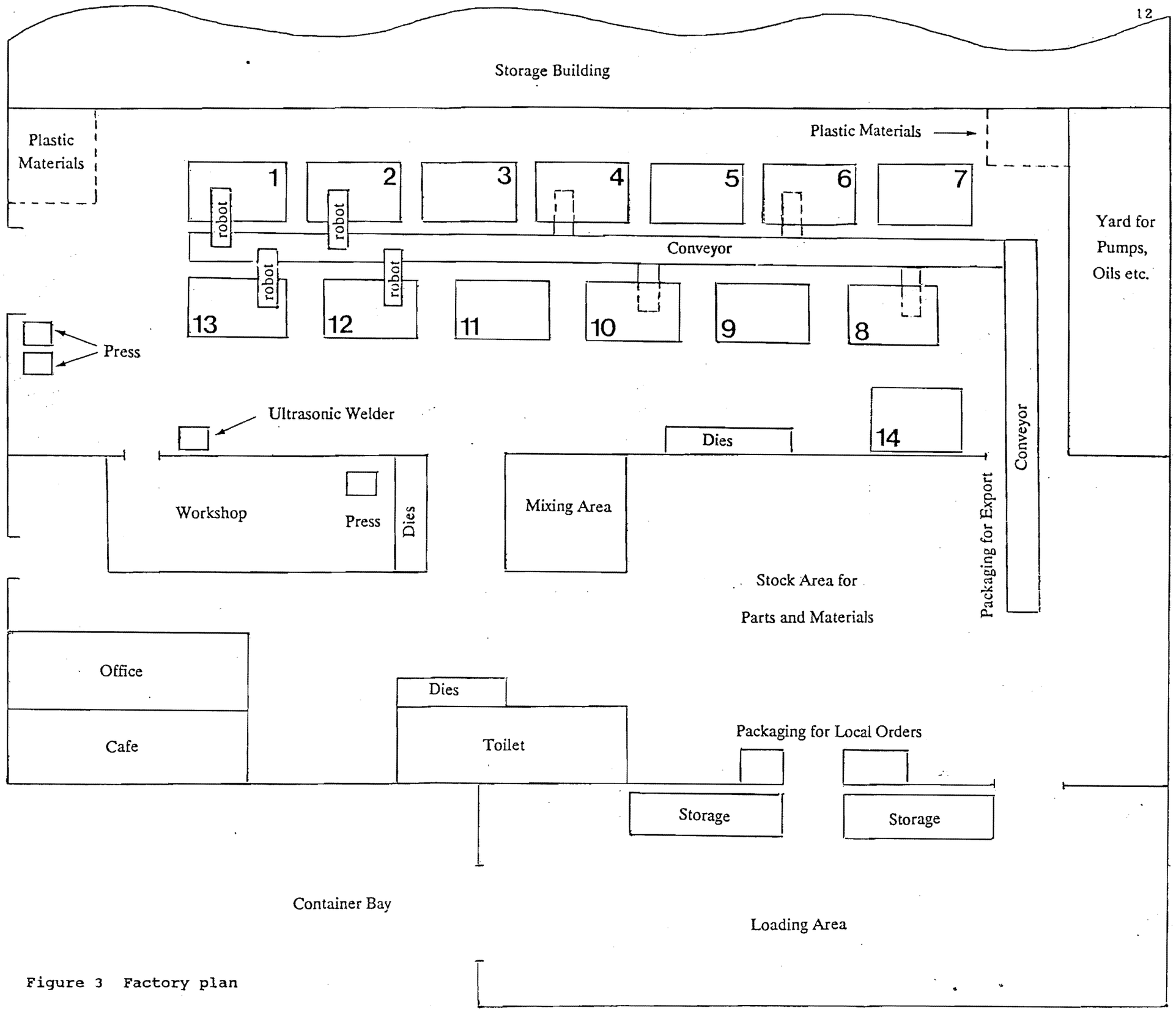


Figure 3 Factory plan

or allowed to be dropped onto conveyors or boxes placed underneath the machines (Figure 3).

The parts were assembled and packed at the end of the conveyor line for large orders, which were usually overseas orders. Some of these parts from the conveyor also went to stock when stock levels were low. Packaging of local orders was done close to the loading area. The parts for local orders mostly came from the stock.

The simple assembly work involved putting two or three parts together. For example, putting a lid onto the canister. For some products, assembly was done on the ultrasonic welder which welded plastic parts together. Some work on non-plastic parts was done on the presses.

Stickers and a bar code were then put on the assembled products. Some products were then put into gift boxes and finally into carton boxes. The carton boxes were moved to the loading area, ready to be loaded onto vans or trucks for local orders or shipping containers for overseas orders.

2.4 Information Flow

Figure 4 shows the information and material flow diagram of the business. For overseas orders, which were transported in shipping containers, the processing

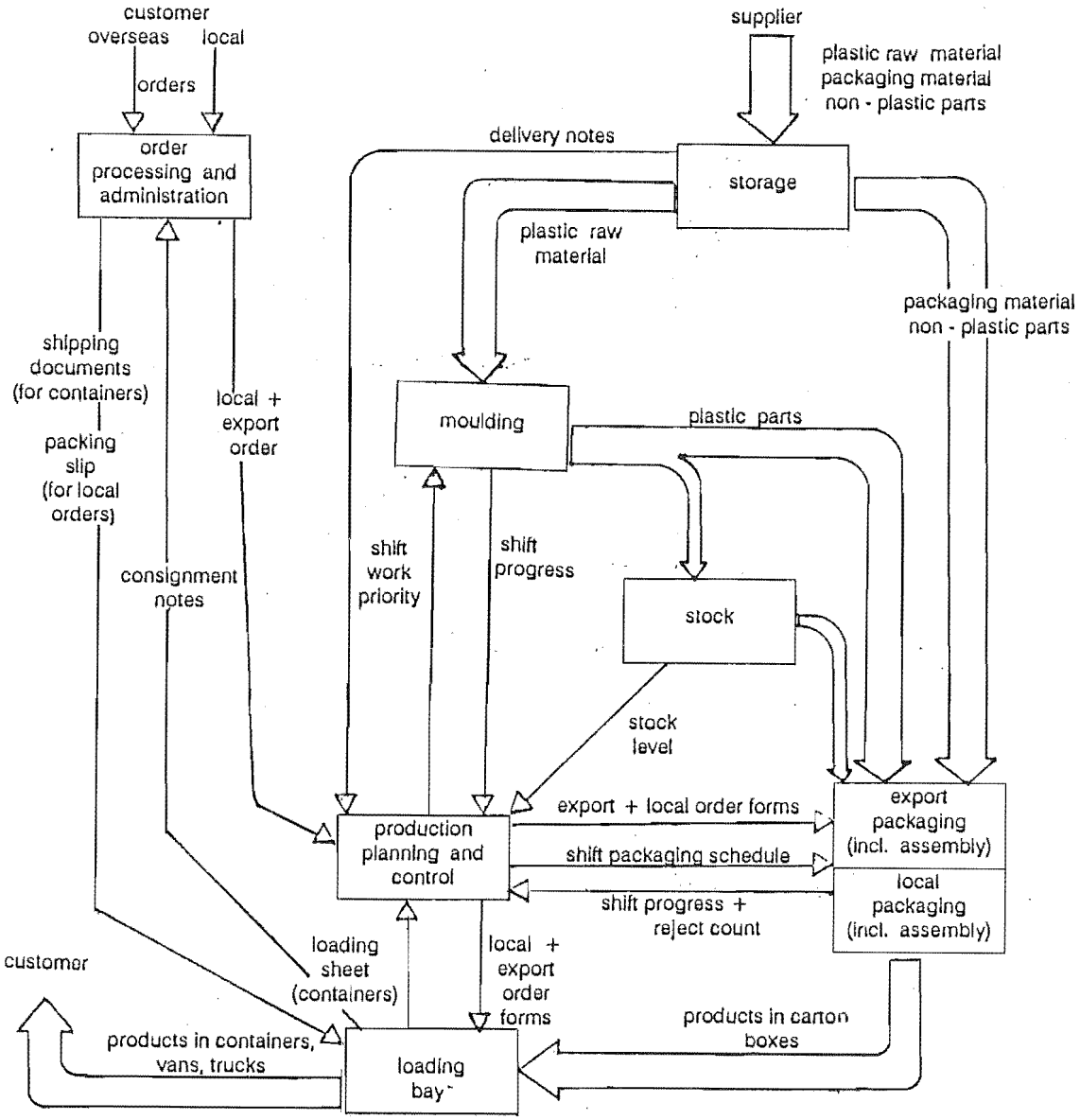


Figure 4 Information and material flow diagram

thin arrow: information flow
bold arrow: material flow

involved calculating the order's fully packed weight and volume. The calculated volume was then compared with several container sizes. If there was space available in the chosen container size, the sales co-ordinator would persuade the customer to order more to fill up the space. The weights of local and overseas orders were quoted on the transport documents.

Next, the shipping schedules were consulted to determine which ship the order would be on to meet the customer's required date. The sales co-ordinator would then pass the export order form to the production manager, detailing the products, their quantity and the date of sailing. The production manager would decide whether or not there was enough factory capacity for that order. If capacity was available, it would be committed to producing for that order. If capacity was lacking, the sales co-ordinator would negotiate a new delivery date with the customer.

For large local orders the same kind of consultation went on between the sales co-ordinator and the factory manager. Small local orders were recorded on local order forms with expected delivery dates and passed on to the factory planners.

At the production planning and scheduling level, a board, with fourteen rows, was used to keep track of the committed orders and the time required on each machine.

As orders came in, the numbers of parts required were calculated and the machine time for each part was estimated. The board was then updated. It was from the information on this board that the machine allocation and packaging schedules were based.

These activities were done for large orders only. For small local orders, the needed parts were taken from stock. Whenever the stock level was low the planners would reserve some machine capacity for local orders.

Packaging capacity was not planned as such. The planners would rely on their experience to estimate the available capacity in that factory. Packaging was a more critical process to control than the moulding process as it was the slower process most of the time. The machines were often run at a slower speed than their capability so that a bottle neck situation would not develop.

For each shift the planners would work out what needed to be produced. Changing of the dies could take less than 15 minutes if everything works out fine but could take longer if problems cropped up. Therefore, a work priority was given to the mould technicians in every shift to fix the more important problems first. Shift progress on what had been produced was reported back to the planners at the end of each shift. There was also a reject count on faulty parts by the packaging team.

In the loading bay, the two storemen who were in charge of this area recorded on the loading sheet, the products that were loaded into the container. On completion of loading each order, the loading sheet would be checked again by the planners. When transport vehicles arrived to pick up the products, consignment forms from the transport company would be sent up to the office to be filled in. Packing slip for local orders or shipping documents for overseas orders would be sent down to the loading bay to be delivered with the orders.

2.5 Deficiencies in the Organisation

From the analysis made of the business, several deficiencies were found in the management and information areas. The more important deficiencies that needed urgent attention are described below:

i) The need to streamline order processing and factory scheduling

There were two time consuming and repetitious processes that could be computerised: the processing of orders and the 'explosion' of products into their parts to estimate the machining time.

The factory management was not satisfied that they had been consulted fully when orders were accepted from customers. The kind of consultations mentioned above sometimes did not happen in busy periods. The managing

director also accepted orders on his overseas sales trips with little consultation with his factory manager. At times the factory manager had difficulty fitting all these orders into the factory schedules. Clearly an information system that would give accurate data on available factory capacity was needed for the factory manager, the sales co-ordinator and the managing director.

ii) Better product costing system

The accountant expressed the opinion that there should be a better product costing system in place. What was available was only material costing for each product. No attempt was made to estimate labour content, factory overhead or administration overhead. When quoting price for an order, rule-of-thumb was sometime used; 'x' times material cost. There was no way of knowing whether those quotes would actually cover the production cost until the accountant has worked out the profit or loss for that period with all the income and expenditure figures.

iii) More efficient way of ordering of materials

The ordering of material was in need of better co-ordination. As mentioned above, the assistant managing director and the people in the factory management did the ordering. The informal arrangements had led to two or more people being responsible for ordering certain items. Much time and trouble had been taken to make sure there

was no multiple ordering of items and items were indeed ordered when stock levels were low.

iv) Stock Keeping

How did the manager knows about the stock levels? He often had to send the storemen to count the materials in stock. A better stock keeping practice needed to be in place. This would make it easier for the yearly stock taking too.

v) No measure of performance

There was virtually no performance indices employed by the management. The managing director was interested only in knowing whether the products were selling well or not. As for the rest of the people in the company they would not know whether they were improving or otherwise in what they were doing. They had little incentives to improve, this was especially true for those in the factory.

There were other areas that were in need of improvement but not as urgently as those mentioned above. The next chapter presents proposals for an information system to solve these problems.

CHAPTER III

ESSENTIAL FEATURES OF AN MRP SYSTEM

The computerisation and production control and planning required by this business could be provided by a material requirement planning system. The repetitious nature of the production made it a potential candidate for the just-in-time (JIT) way of planning and control, which would bring in a lot of benefits. This was not considered as JIT requires radical changes in the organisation, with which this business would not be able to cope.

JIT advocate, Edward Hay wrote:

> If a company is coming apart at the seams, my inclination is to say to implement MRP, both in order to get control and in order to keep the situation from deteriorating further by asking employees to make such a radical change as JIT [3].

This chapter looks at the essential features of an MRP system and shows how some of the features have been altered to suit the conditions in this business and for ease of computation.

3.1 Types of MRP System

An MRP system can be used in a variety of different ways. Schroeder has summarised them as three different types described as follows [4]:

Type I: An inventory control system.
It is an inventory control system which releases manufacturing and purchase orders for the right quantities at the right time to support the master schedule. It does not include capacity planning.

Type II: A production and inventory control system.
It is an information system used to plan and control inventories and capacities in manufacturing companies. In this system, the orders resulting from parts explosion are checked to see whether sufficient capacity is available. If there is not enough capacity, either the capacity or the master schedule is changed. This system has a feedback loop between the orders launched and the master schedule to adjust for capacity availability. As a result this type of MRP system is called a closed-loop system; it controls both inventories and capacity. This is shown in figure 5.

Type III: A manufacturing resource-planning system.
The type III MRP system is used to plan and control all manufacturing resources: inventory, capacity, cash,

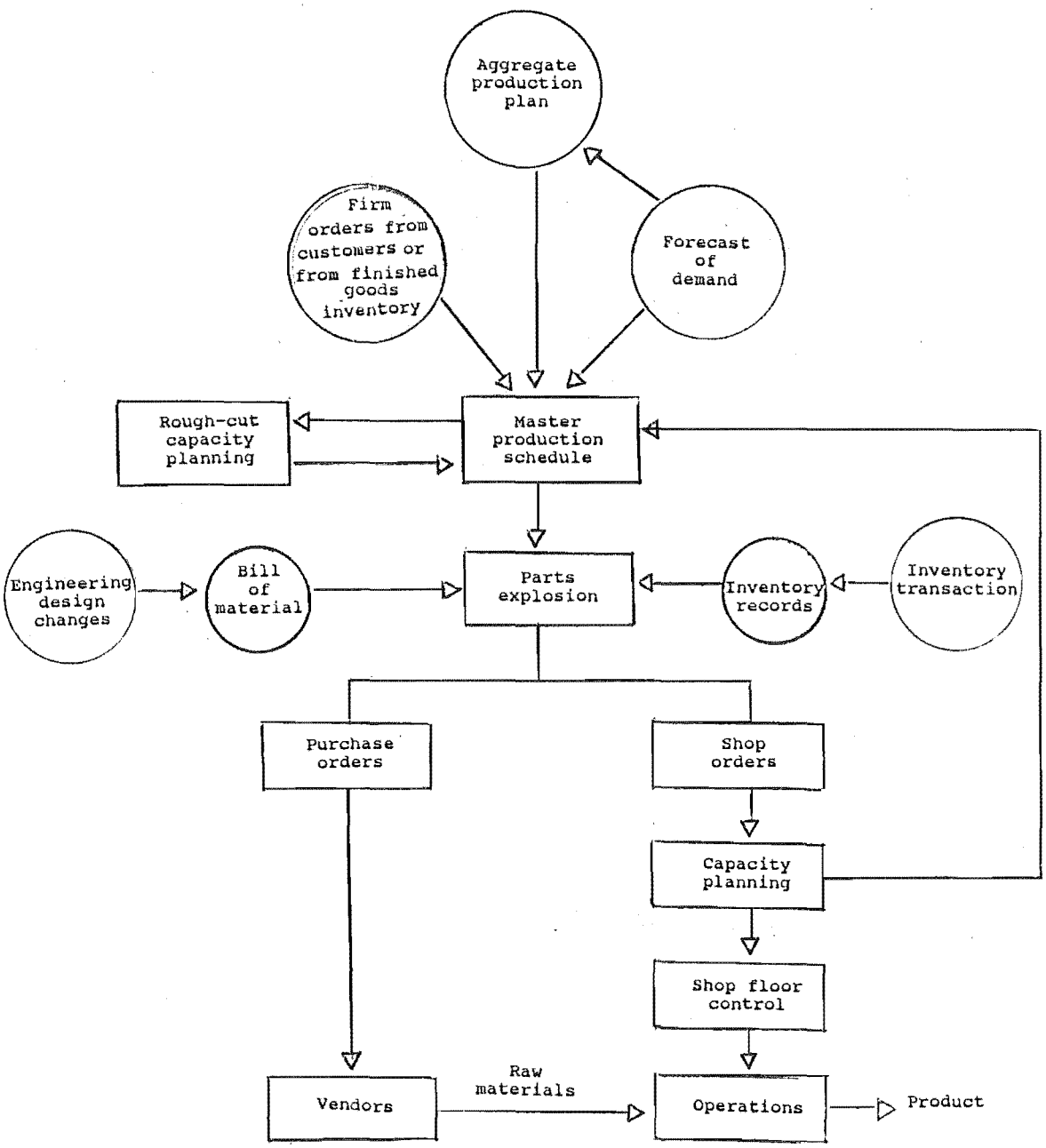


Figure 5. Closed-loop MRP system [4]

personnel, facilities, and capital equipment. In this case the MRP parts-explosion system also drives all other resource-planning subsystems in the company.

The MRP system that this project is concerned with is perhaps best described by the MRP type II system. It is not expected that all MRP systems should fall neatly into any one of these three types of systems. All information systems should reflect the needs of the business, and these needs differ from one business to the others. Despite these differences, a successful MRP system would have to incorporate a few essential features which are described below.

3.2 Essential Features

3.2.1 Master production schedule (MPS)

An MRP system is driven by the MPS which specifies the "end items" or output of the production function. All future demands for work-in-process should be dependent on the master schedule and the material requirements should also be derived from the master schedule.

The top of figure 5 shows information from three sources that is needed for scheduling. The aggregate production plan deals with families of products or product lines and is usually part of the annual budgeting

and strategic planning process. Aggregate production planning and forecasting of demand are outside the scope of this project. It is assumed that information from these two sources are on hand when scheduling is done.

Scheduling is often done on a weekly basis. The MPS consists of weekly time buckets. In the case of this system the time buckets are further separated into a large orders section and a small orders section. This is to ensure that capacity in any single week is not overwhelmed by large orders and that small orders are catered for. The period covered by the master schedule should be long enough to ensure sufficient time is available to order all parts and materials.

It is very important that the MPS reflects realistic capacity constraints. So before the MPS is to be used, a capacity planning routine must be run to determine whether sufficient machine capacity, material and human resources are available. The methods employed to do this are explained later.

3.2.2 Bills of materials (BOM)

To be able to derive material and capacity requirements from the master schedule, an accurate BOM is needed. BOM is a structured list of all the materials or parts needed to produce a particular product, assembly, subassembly, manufactured or purchased part.

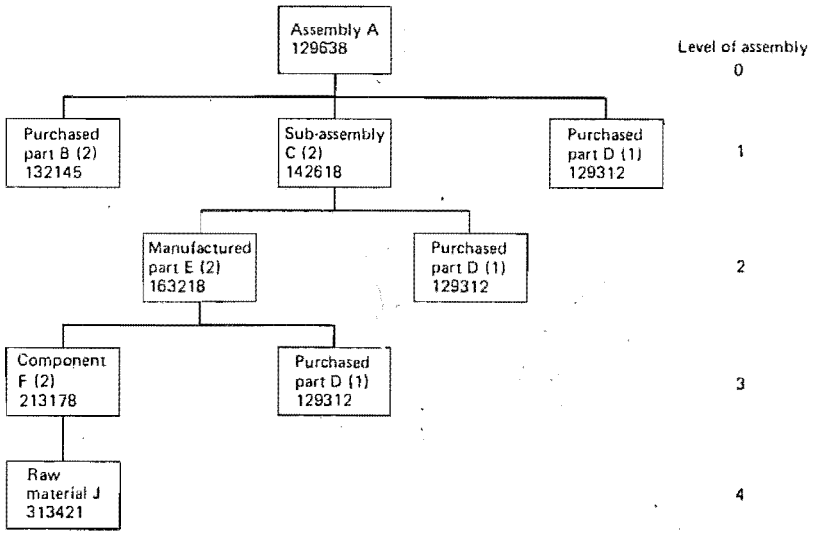


Figure 6. Product structure tree example [5]

(Key: (2) required units for higher level assembly
132145 inventory number)

Figure 6 shows the type of information in a typical product structure tree, which the BOM would have to hold for all products: it lists all components of a product and how they are related. A prerequisite to setting up a BOM is that each inventory item is identified by a unique code (inventory number).

3.2.3 Capacity planning

There are two ways to check the validity of the MPS:

1. Rough-cut capacity planning (or resource planning)
Approximate labour hours and machine hours are calculated directly from the master schedule to project future capacity needs without going through the parts-explosion

process. When sufficient capacity is not available, the master schedule is adjusted or capacity is changed to obtain a feasible schedule. Only then is the parts-explosion run, to get the materials requirements.

2. Shop loading

A full parts-explosion is run at the beginning to get the resulting shop orders. These are loaded against work centres through the use of detailed parts-routing data. This gives a projected future capacity requirement for each work centre. As in the first method, the capacity or the MPS is adjusted until the MPS becomes feasible. At this point a valid material plan is available.

Rough-cut planning involves less calculations but is not as accurate as shop loading. Either one or both of these methods can be used, depending on the system requirements.

The products produced by the factory being studied had a simple product structure. As the parts-explosion did not involve much computation, the more accurate technique was chosen.

3.2.4 Computation of material requirements

The objective of material requirements computation is to determine the action to be taken on the inventory items. The course of actions to be taken with regard to

each inventory item is in turn guided by the following questions:

What do we have?

What do we need?

What do we do?

Some basic elements of the inventory status that are needed to answer the above questions include:

Quantities on hand

Quantities on order

Gross requirement quantities

Net requirement quantities

Planned-order quantities

On-hand and on-order quantities are reported to the system and can be verified readily by inspection. The other three quantities are computed by the system and can be verified only through recomputation. They are explained below.

The gross requirement of an inventory item is defined as the quantity that will be issued to support a parent order (or orders). For example, the gross requirement of a plastic material derives from all the plastic parts that will be moulded from that material.

The computation of the net requirement is as follows:

	Gross requirement
minus	on-order quantity
minus	on-hand quantity
equals	net requirement

The planned order quantity of an item is equal to its net requirement. The release of planned order is arrived at by offsetting for lead time.

One important factor that also needs to be considered in the computation of material requirements is the safety stock. The primary purpose of safety stock is to compensate for fluctuation in demand, i.e. for forecast error. Therefore, safety stock, where required, should be considered at the master schedule level. This has the advantage of ensuring that matching sets of components, not simply an assortment of various parts, are available for production. The secondary function of safety stock is to compensate for uncertainty of supply. For items which are known to have erratic resupply performance, it is justifiable to allow for safety stock.

Two types of stocks may be maintained in the factory: purchased materials and purchased parts. When the supplier of a particular material is known to be unreliable, safety stock is then allowed for. The inventory records have adjustable maximum, mean and

minimum stock levels set by the management. The aim of the system is to maintain the material stock levels within a range as close to the mean level as possible. In the long run the management should aim to eliminate the need for safety stock, that is to bring the mean level down to zero. This can be achieved by negotiating and working with the supplier to ensure a more reliable supply.

The stock for parts is maintained mainly as a buffer stock between the moulding and packaging processes. The dies may give trouble from time to time and the stock could provide the continuity of parts to the packaging team.

CHAPTER IV

MODELLING THE SYSTEM

Modelling the system is an essential stage in the system development. The system model should be able to provide the necessary information, which is needed to build the database and end-user applications. It should show the data requirements and the activities (processes) of the system. Normally, it will require more than one type of modelling technique to do this.

In this project, three diagramming techniques were employed to model the system. They were chosen because they were able to provide the required information * and they had the ability to support the top-down approach.

4.1 Decomposition Diagrams

The first and the simplest of the diagramming techniques is the decomposition diagramming. A high-level organisation, function, or activity is decomposed into

---

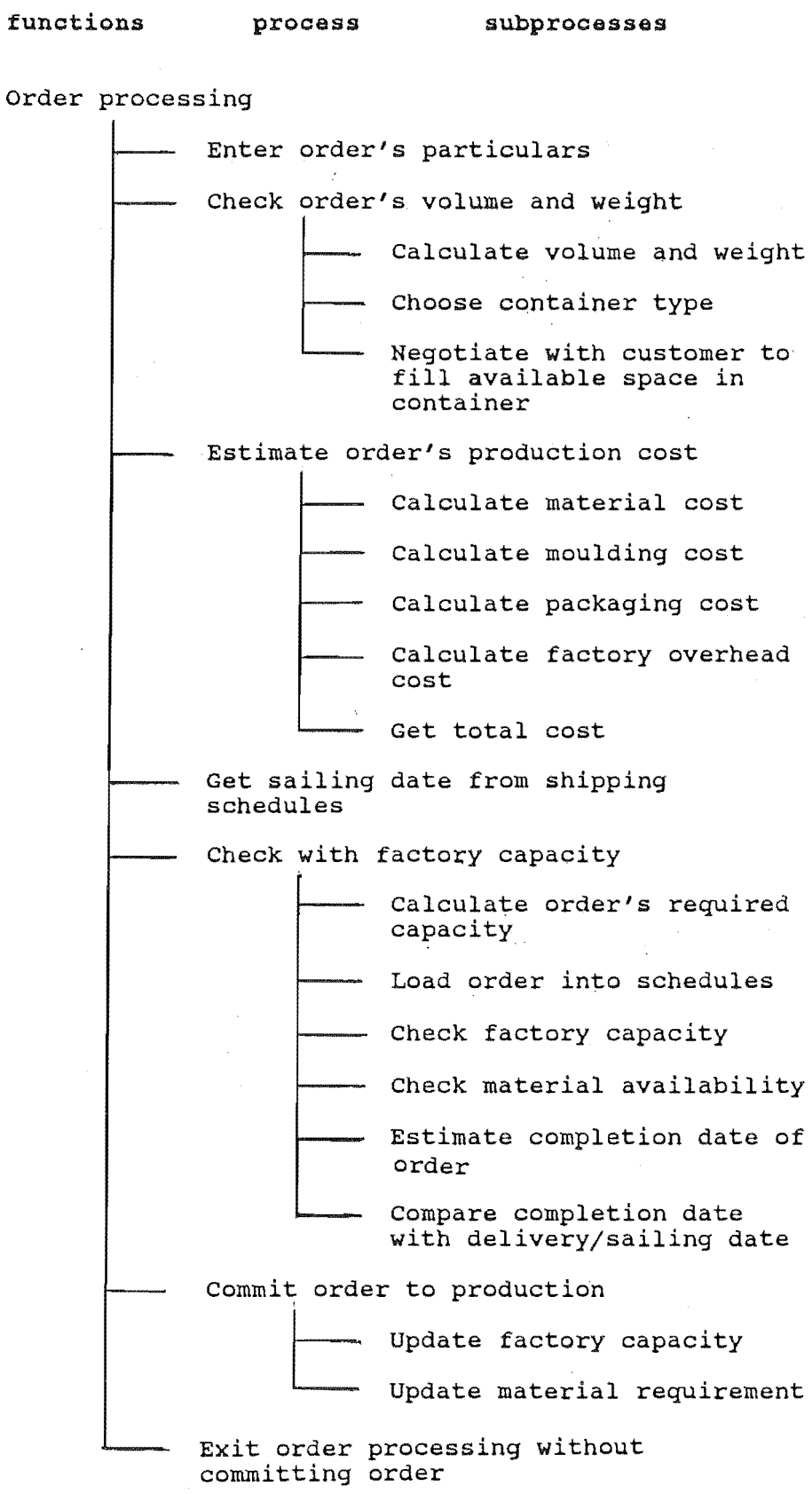
* Cronin demonstrated in his book, Mastering Oracle, that these three diagramming techniques provided sufficient information to build a relational database and applications, by using the Oracle software tools (Daniel J. Cronin, Mastering Oracle, 1988). The same software tools were used in this project.

lower-level activities or processes. The lower in the hierarchy, the greater will be the details.

Decomposition at the highest level is termed functional decomposition. It illustrates the major functions or activities of the corporation.

Lower down in the hierarchy are the processes and sub-processes, consist of mainly specific activities that have to be carried out.

The functions of the organisation under study were summarised into two major areas: order processing and production planning and control (see section 2.5). The decomposition diagrams for this system are shown below.


functions
process
subprocesses
Order processing
Enter order's particulars
Check order's volume and weight
Calculate volume and weight
Choose container type
Negotiate with customer to fill available space in container
Estimate order's production cost
Calculate material cost
Calculate moulding cost
Calculate packaging cost
Calculate factory overhead cost
Get total cost
Get sailing date from shipping schedules
Check with factory capacity
Calculate order's required capacity
Load order into schedules
Check factory capacity
Check material availability
Estimate completion date of order
Compare completion date with delivery/sailing date
Commit order to production
Update factory capacity
Update material requirement
Exit order processing without committing order

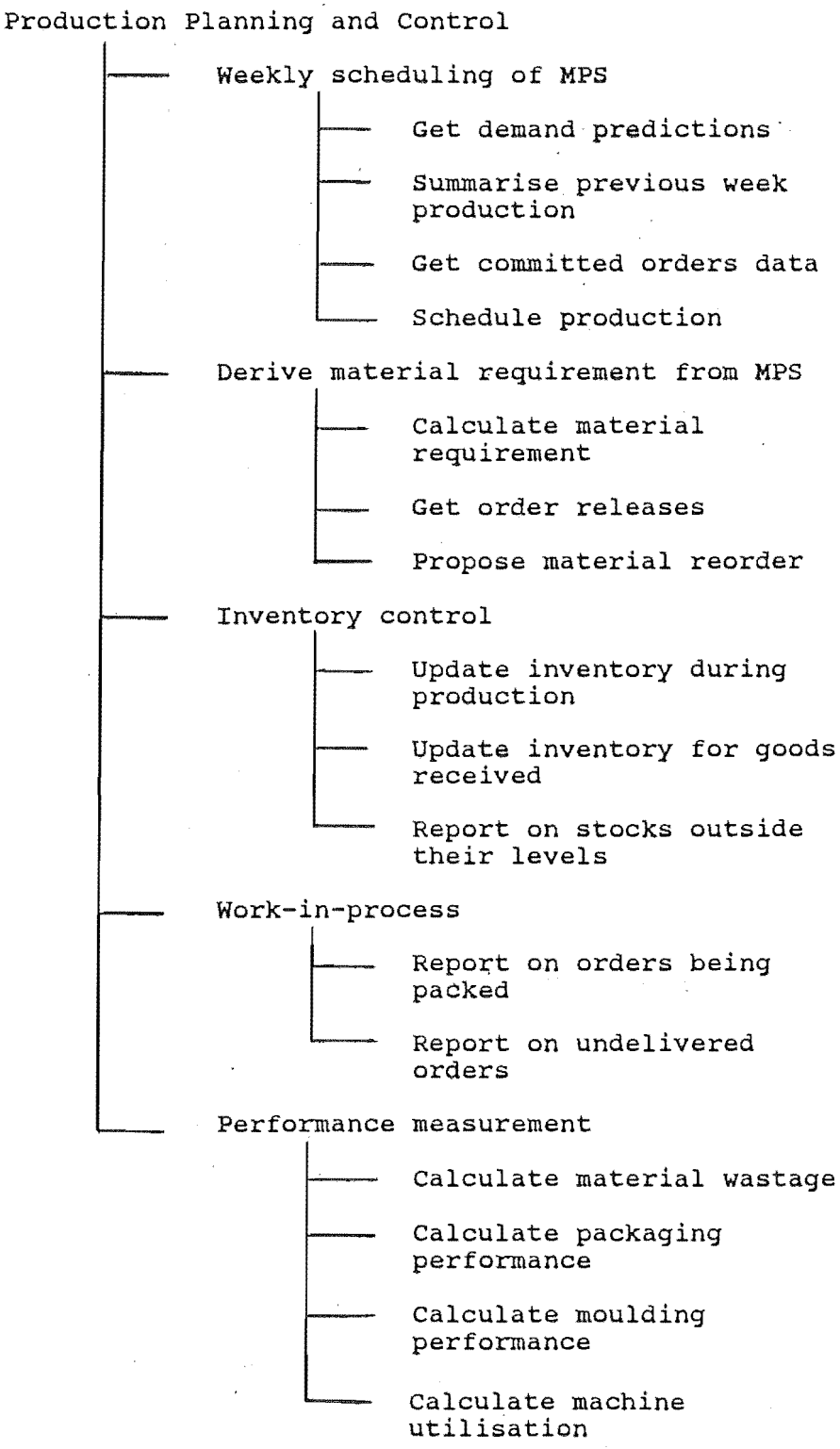


Figure 7 Decomposition diagrams

4.2 Entity-relationship Diagrams

If the decomposition diagram is to be taken as one half of the equation in modelling the system, the entity-relationship diagram is the other half. An entity is something (real or abstract) about which data is stored. Some examples of entities in a business are clients, suppliers, divisions or departments. The entity-relationship diagram assists the designer in defining and understanding the significant entities in the business and also the relationships between these entities.

The entity-relationship diagram of the system is shown in figure 8. The keys to this diagram are given in appendix A.

In building an entity model, the simplest core relationships are usually defined first. Other relationships are then added, layer by layer to make a complete model.

The core relationships in the system were the factory capacity-MPS-MRP relationship and the product-unit-part relationship. The latter relationship is explained below.

A unit was defined as an assembled item, consisting of two or more parts; a product contained a number of different units, mixed and packed in a specific way. For example, six canisters with gift boxes might be considered as one

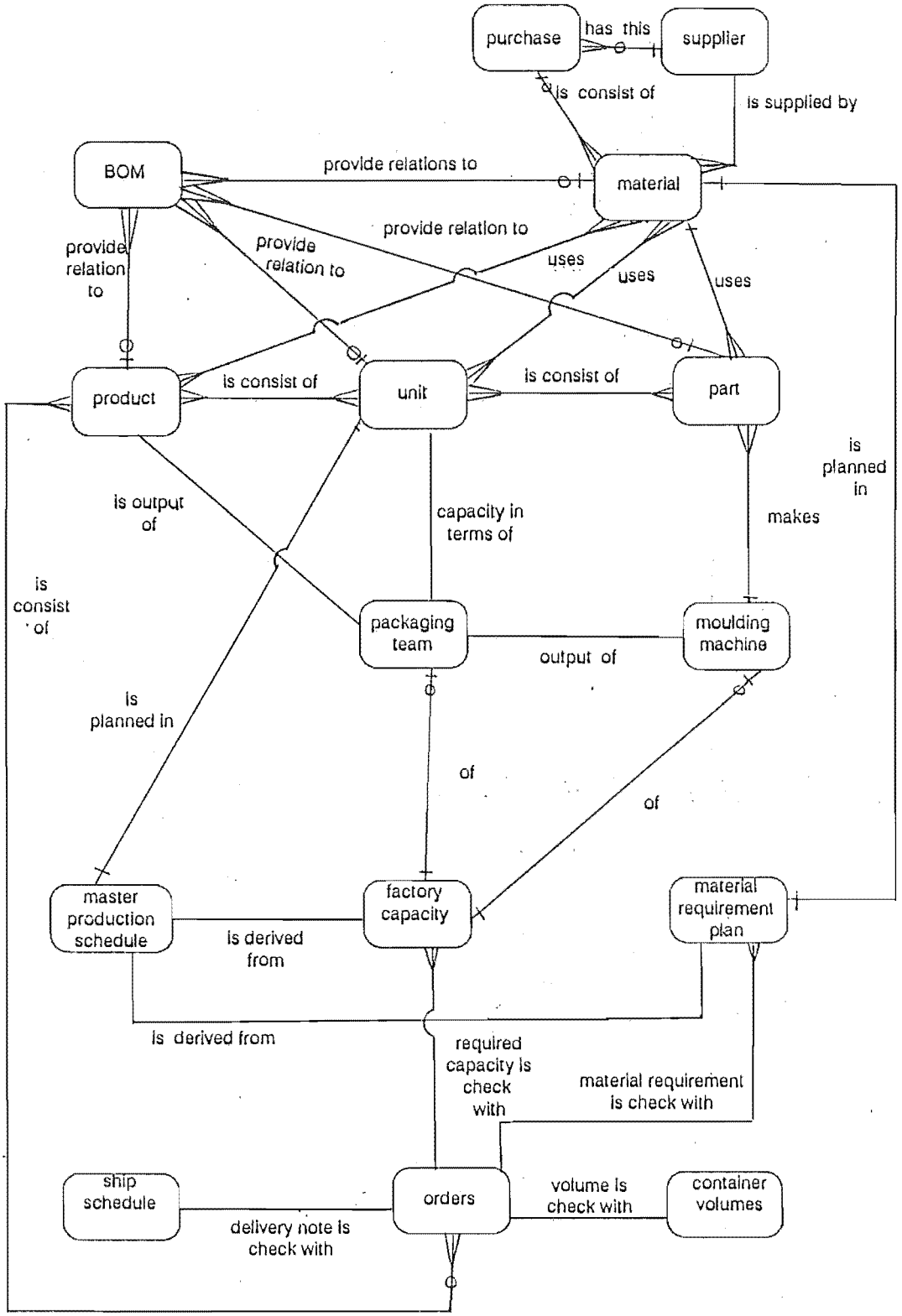


Figure 8 Entity-relationship diagram

product, while two canisters and two lunch boxes without gift boxes might form another.

In this way, the number of different products was in the hundreds. This would make production planning (in the MPS) in term of products difficult to manage. The number of different units was less than sixty, a more manageable number. Therefore, it was decided that the planning would be done in term of units. However, it recognised that the marketing and long term planning should be carried out in relation to products or product lines.

The other entities and relationships were defined and added on to the core relationships, as shown in figure 8.

4.3 Data Flow Diagram (DFD)

The data flow diagram (DFD) binds the business functions and entities, providing a graphic representation of their interrelatedness. It captures the input and output flow of data in the entire system. It also serves as a platform for designing the database, menu and user interface.

Again, by taking a top-down approach, the overview DFD of the system was worked out first (figure 9). The diagram showed only the source (the origin of data), and the sink (the ultimate recipient of data). The DFDs for all the processes were then detailed out from this overview diagram. The detail diagrams are given in appendix B.

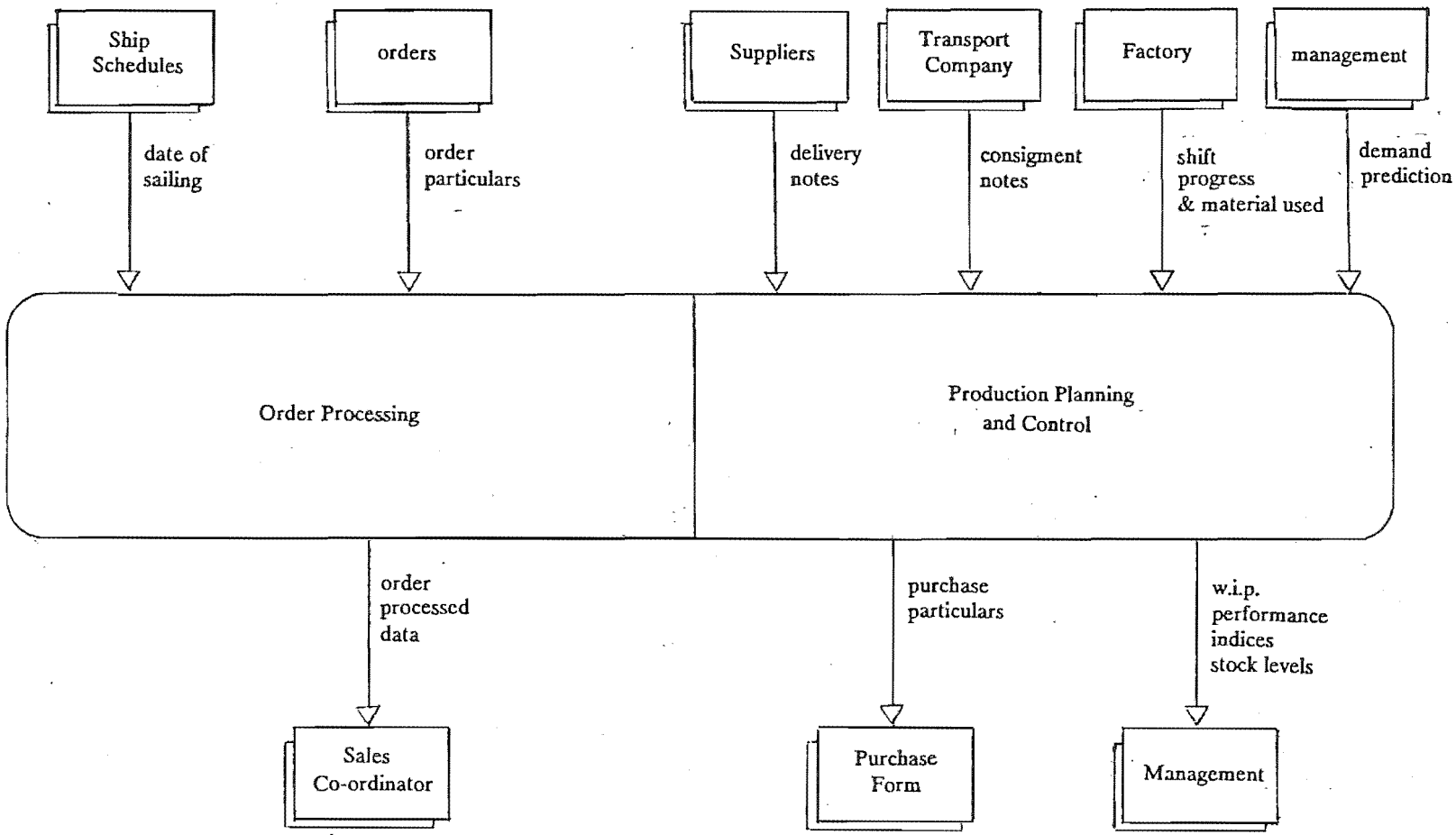


Figure 9 Data flow diagram

In detailing the diagrams, it was necessary to establish the types of information that have to be kept in each file. This information was particularly important when designing the database. It also became clearer at this stage which of the files or processes would be computerised. Some of them might be more efficient to be used in the non-computerised form.

One such file was the shipping schedules. The amount of manual data entry for such schedules was considerable and it required updating regularly (see example in appendix C). Unless the data was kept electronically in the shipping companies and could be accessed by or down loaded into the user computer, the computerised schedules might not be efficient to use. The possibility of electronically linking up the shipping schedules was not investigated in this project and they were left uncomputerised.

CHAPTER V

INTRODUCTION TO RELATIONAL DATABASE, 4GL AND ORACLE SOFTWARE PACKAGE

This chapter explains the relational database approach to data organisation in a computer system and the type of software technology that was chosen. It also introduces the Oracle software tools that were used to build the system.

5.1 Relational Database

The application-centred approach is one way of organising the data in a system (figure 10). In this approach, data files used by applications are linked to individual applications only. Transfer files (T1-T5) facilitate the communication between the applications. However, this approach has some serious weakness. It can easily bring in problems of inconsistency in the data and unnecessary duplications of data.

The database approach will greatly reduce the above problems. Data is stored as a common pool to be shared between applications (figure 11). Knowing the overall requirements of the enterprise - as opposed to the requirements of any individual user - the database system can be structured to provide an overall service

that is "best for the enterprise." This is very much in line with the system approach.

Security can be a problem with the sharing of data. Therefore, the database system must be structured such that (a) the only means of access to the database is through proper channels, and (b) authorisation checks will be invoked whenever sensitive data is accessed.

What then is a relational database? Relational database holds all information about objects and the relationships among those objects in the form of tables. All the data in a relational database is integrated and shared. Integration ensures data is logically efficient, thus eliminating unnecessary duplication and providing easy data access. Also, data presented in tables is more easily understood by non-specialists.

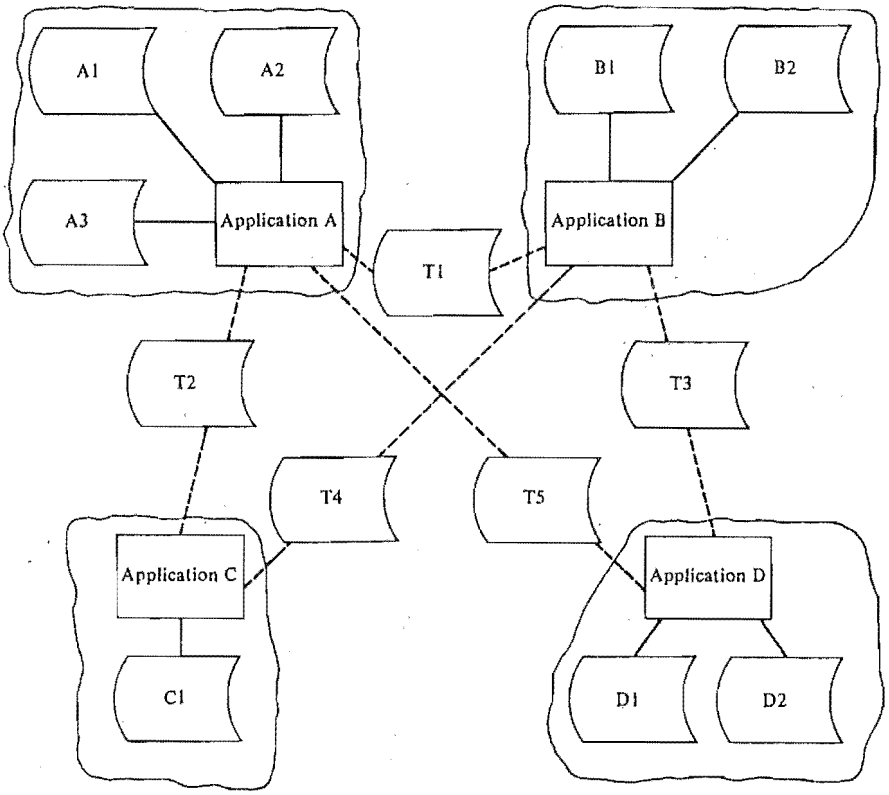


Figure 10 Application-centred data organisation [8]

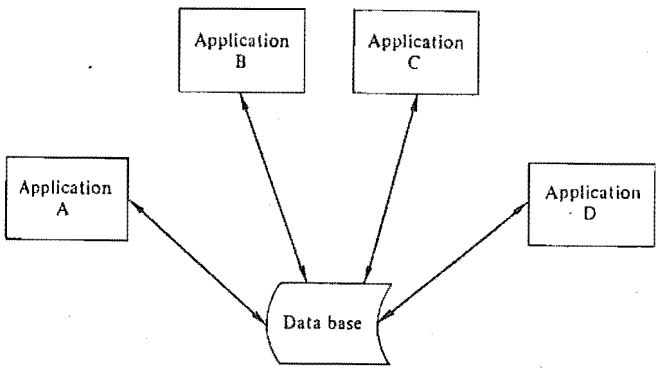


Figure 11 Database approach [8]

Tables in a relational database have these attributes:

1. Each row (each record) is unique, there can be no duplicate rows.
2. Each entry in table represents a single data item, there are no repeating groups.
3. Every row has exactly the same structure. (Each record has the same fields of the same size in the same order).
4. Each data item has a single value.
5. Each column is assigned a unique name.
6. All the values in a column are values of the same attribute, that is they all come from one predefined set of possible values.
7. The order of the columns is immaterial.
8. The order of the rows is immaterial.
9. Both rows and columns can be viewed in any sequence at any time without affecting their information content.

Figure 12 Relational table attributes [9]

The normalisation process, which the designer can use to design such tables will be explained in the next chapter.

5.2 Fourth Generation Language (4GL)

There is a wide range of software packages available in the market. Selecting one to suit the applications is quite a task. A guide in choosing the type of software technology to use is shown in figure 13.

Earlier in the project, an attempt was made to use the company's spreadsheet software package called LOGISTIX. A database and macros were developed to calculate weight, volume, material cost and machining time needed for orders. The hardware used was ICL PC model 46, with 20 Mbytes fixed disc and 551 Kbytes user RAM. The spreadsheet took about 10 minutes to process a simple order.

Though with a more sophisticated spreadsheet and a faster machine the processing time could be shortened, the experience here showed that a spreadsheet would not be able to handle an MRP system efficiently. Processes such as parts-explosion, calculation of capacities and material requirements, and processing of orders require a lot of data manipulation. It would take a database management system (DBMS) and a database access and manipulation language to do this efficiently. DBMS acts as a software interface that does the translation between the global view of the data in the database and the local view expected by each application program.

VERY SMALL APPLICATIONS - <500 LINES OF SOURCE CODE

- o SPREADSHEETS
- o INTERPRETED LANGUAGES (BASIC, C)

SMALL APPLICATIONS - 500-2000 LINES

- o END USER FOURTH GENERATION LANGUAGES

LOW-MEDIUM APPLICATIONS - 2000-8000 LINES

- o PRODUCTION FOURTH GENERATION LANGUAGES
- o PROGRAM AND APPLICATION GENERATORS

MEDIUM APPLICATIONS - 8000-32,000 LINES

- o CASE DESIGN TOOLKITS
- o FOURTH GENERATION LANGUAGES
- o PROGRAM AND APPLICATION GENERATORS
- o THIRD GENERATION LANGUAGES

MEDIUM-HIGH APPLICATIONS - 32,000-128,000 LINES

- o CASE ANALYSIS AND DESIGN TOOLKITS
- o PROTOTYPING TOOLS
- o THIRD GENERATION LANGUAGES

LARGE SYSTEMS - 128,000-512,000 LINES

- o CASE ANALYST AND DESIGN TOOLKITS
- o PROTOTYPING TOOLS
- o SECOND AND THIRD GENERATION LANGUAGES
- o REUSABILITY

VERY-LARGE SYSTEMS - 512,000-2 MILLION LINES

- o ANALYSIS, DESIGN, PROJECT MANAGEMENT TOOLKITS
- o REUSABILITY
- o DEVELOPMENT METHODOLOGY
- o SECOND AND THIRD GENERATION LANGUAGES

Figure 13 Application size to software technology match [10]

THIRD GENERATION TOOLS

- COMPILERS
- PROGRAM EDITORS & LIBRARIES
- TEST DATA GENERATORS & COMPARATORS
- PERFORMANCE MONITORS & OPTIMIZERS

FOURTH GENERATION TOOLS

- FOURTH GENERATION LANGUAGES
- DBMS
- APPLICATION GENERATORS & CODE GENERATORS

FIFTH GENERATION TOOLS

- NATURAL LANGUAGE INTERREPTTERS
- SPEECH RECOGNIZER
- VOICE SYNTHESIZER
- PARALLEL PROCESSOR

Figure 14 Computer software generations [10]

DBMS is a tool associated with the 4GLs (see figure 14). What is a 4GL? There is no standard definition for a 4GL tool or system. Features of a 4GL include ease of use, less instructions and time are required in application development compared with third-generation languages such as COBOL or PL/1, and suitable for nontechnical end-users. Many characterise a 4GL system as one where a programmer specifies the task to be achieved, while leaving the knowledge of how to achieve the task to the language itself [11].

A 4GL system should support:

* high-level user functions such as screen formatting, menu and logical device management,
* data-oriented functions such as structuring of data, storage management, and
* system functions such as file handling and communication with other applications.

4GL is the favoured software technology for medium to small applications. As the system in this project was not large or too complicated and because of the user-friendliness of the softwares and the need for a DBMS, 4GL was chosen as the software platform to build the system.

5.3 ORACLE Products

A list of the more popular 4GL softwares is given in appendix D. A 4GL software package called ORACLE, was used as the system development software because of the following:

* It is a relational database system and therefore has the advantages mentioned earlier.

* It uses the industry-standard relational database language, SQL. This means that future development work can also access a wide range of SQL-based products and is less likely to become an obsolete language in the fast changing computer world.

* Applications developed on Oracle can be ported to over 80 hardware platforms with virtually no modifications required. This gives great flexibility to future development work.

* By using default settings provided in its application generator, a skeleton of the system can be developed quickly.

* Oracle offers good database security. Access check can be designed at applications, table, column or row levels.

* As one of the world's top selling database systems it is a proven product.

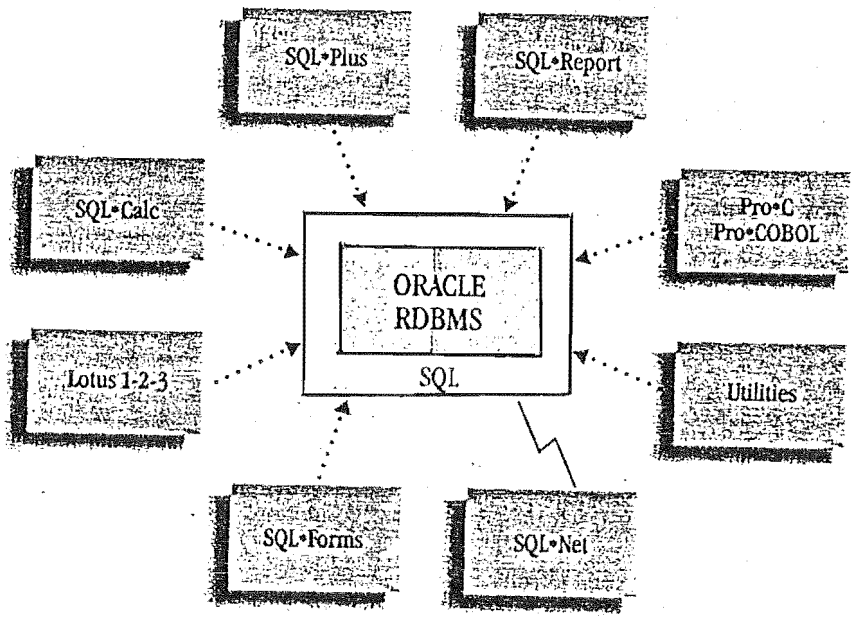


Figure 15 Oracle software products [12]

The Oracle products for personal computers, as illustrated in figure 15, consist of:

Oracle Relational Database Management System (RDBMS)
It manages the data storage environment and the access to data. The data base with its associated tables are stored in an active data dictionary. This means that the system automatically updates the dictionary in "real time" as changes are made.

Structured Query Language (SQL)

SQL is the interface language between the user and the Oracle database. Developed by the IBM corporation, SQL was accepted by the American National Standards Institute (ANSI) in 1986 as the standard access language of RDBMS.

A set of SQL-based tools and utilities is available to help users in developing and running applications. Three of these tools, namely SQL*Plus, SQL*Forms and SQL*Menu, were used to develop the system in this project. The essential features of these tools are described below.

SQL*Plus

Of all the tools, SQL*Plus provides the most direct access to data. SQL*Plus includes the full standard SQL commands, plus the Oracle developed extensions to SQL, called SQL*Plus commands. With these commands, users can then perform ad hoc queries and data base administration functions or run pre-defined reports.

SQL*Forms

Applications are mostly built on forms. These forms serve as an interface between the users and the system that has been built. Users can quickly insert, update, delete or query records in the corresponding database tables. These functions can also be performed in

SQL*Plus, but it would be laboriously slow if there are more than a few records to be manipulated.

Functions that can be designed in the forms for the control of data entry are:

* provision of default values;
* validation of data to ensure it is correct;
* cursor navigation to control user movement through the application;
* automatic display of related data; and
* automatic computation of field values.

At the heart of SQL*Forms are the triggers. They are strings of SQL statements or SQL*Forms commands placed at various event points in the form. When a certain event occurs, a predefined trigger will fire, executing a set of instructions.

SQL*Menu

SQL*Menu is a management tool for linking all the applications, even non-Oracle applications. Access to the applications is through a set of options presented in a traditional menu style. It can be designed to designate authorisation levels, so users can only access those options which they are authorised to use.

The hardware requirements for using Oracle are given in appendix E.

CHAPTER VI

BUILDING THE SYSTEM

6.1 Building the Database

In building the database, all information or attributes associated with each entity/files would have to be gathered. The data flow diagrams were able to provide this information. These attributes and entities formed the columns and tables of the database respectively. To transform the tables to a relational database, the normalisation process, described below, was followed.

Normalisation simplified each table to its primary key and attributes. The goal of normalisation was to minimise data redundancy and to allow data to be easily managed and changed.

6.1.1 Primary and foreign keys

A primary key is an element that uniquely identifies each row of a table. The primary key can be stored in one column or in a combination of columns.

To link two or more related tables, the primary key of one table is stored in the other table(s) as a reference. The key is then called a foreign key in the latter table(s).

The following illustrates an example from this project:

PURCHASE table

column	attribute	key
purno	purchase number	primary (P)
supno	supplier number	foreign (F)
datesent	date purchase form was sent	
daterec	date goods were received	
matno	material number	P,F
qty	quantity ordered	
received	quantity received	

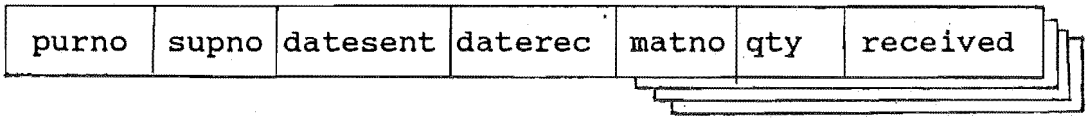
purno	supno	datesent	daterec	matno	qty	received
101	123	4-12-90	8-12-90	601	50	50
101	123	4-12-90	8-12-90	602	70	69
102	111	1-01-91	5-01-91	601	80	80

Each purchase of any one type of material from a supplier has a unique number. The combination of purno and matno is unique to each row in the table and therefore functions as a primary key. Matno and supno are themselves primary keys in the material and supplier tables respectively, but are foreign keys in the purchase table.

6.1.2 Flat table

The next step is to look for repeating group in the data. In the example above, every purchase has one or multiple set of matno, qty and received values. They are

the repeating group in the PURCHASE table. As a result, the table is called a 'non-flat table' (see illustration below).



For sound database design, all tables must be reduced to two dimension or flat tables. The above PURCHASE table should thus be converted into two separate flat tables:

PURCHASE table		PUR_SUP table	
column	key	column	key
purno	P	purno	P,F
matno	P,F	supno	P,F
qty		datesent	
received		daterec	

Tables that are flat are said to be in a normal form. The tables in this project had all been converted into normal form. (For details of the different degrees of normal forms, please see Appendix G).

6.1.3 Duplicated data

Duplicated data is present when a column has two or more identical values or when the same data is stored in more than one table. The latter case is allowable only if the data is acting as a foreign key.

Within the column, there should not be any redundant data. This is data that can be deleted without information being lost. Redundancy is unnecessary duplication. All tables should be checked for this redundancy.

For example, in the PURCHASE table,

purno	matno	qty	received
101	601	50	50
101	602	70	69
102	601	80	80

Purno value of 101 and matno value of 601 are necessary duplication. Deletion of any of these values will mean that the purno-matno relationship will not be complete. There is no redundancy in this table.

6.1.4 Normalisation processes

The finalised columns and tables are shown in appendix H. Some columns, like the calculation columns, were added to the tables for computation purposes, at the SQL*Forms designing stage. More normalisation processes are described below to show the construction of some of the tables.

The many-to-many relationships of product-unit, unit-part, product-material, unit-material and part-material were normalised by having the BOM table. It kept data of the relationships described above and nothing else. The invno

and have columns in the BOM table were primary keys as well as foreign keys. They represented respectively the "parent" and "child" in a relationship. For example,

BOM table

INVNO	HAVE	QTY
19533	9500	4
19533	9505	3

This meant, inventory 19533 have 4 of inventory 9500 and 3 of inventory 9505.

In the unit-MPS relationship, unit number was the primary key to both tables of unit and MPS. The unit table had the unit number column and attributes associated with the units (for example, unit's name, packaging time for unit, etc.). The MPS table had the unit number column and columns representing the weekly buckets of planned and committed quantities of unit. Clearly the unit number was a redundant column. To remove the redundant column, the unit and MPS tables were merged into a larger MPS table.

The same was true with the factory capacity-packaging-moulding machine relationship (see figure 8). The factory capacity had process number as its primary key. The process number uniquely identified the packaging processes (local and overseas packaging), and the moulding processes (the fourteen moulding machines).

For instance in the factory capacity table, the following information was provided :

Process	Name	Week1 Week2 Week3....
90	Packaging-local	
91	Packaging-o.s.	
1	Machine # 1	
2	Machine # 2	
.	.	
.	.	

The columns week1, week2, week3 etc. were where the available capacity for the week was kept.

The process number (a primary key) was repeated in the packaging and moulding machine tables. To reduce redundancy the three tables, factory capacity, packaging and moulding machine, were merged to form the capacity table.

6.1.5 SQL*Plus

Tables were constructed by using commands in SQL*Plus. The CREATE TABLE command was used to specify

- the name of the table,
- the name of each column,
- the format of each column, which includes
  - the type of data stored (number, character or date),
  - the column width, and
  - whether or not the column must have a value in it.

For example,

```
CREATE TABLE pur_sup
   (purno number(6) not null,
    supno number(3) not null,
    datesent date,
    daterec date)
```

New columns could be added to a table easily by using the ALTER TABLE command. Deletion of column was also easy but not as straightforward. The CREATE TABLE, SELECT, DROP and RENAME commands were needed to do the job. With these commands, there was great flexibility to alter the structure of the database, no matter how much data had already been stored. With a good relational design, additional modules could be integrated easily into the database.

After creating the tables, data could be inserted using the INSERT command. With new data, this command allowed only a row of data to be inserted. Multiple rows of new

data required multiple INSERT commands. A faster way was to use the SQL*Forms default settings to create fields corresponding to the columns in the table and insert data through the form.

6.2 Building the User Interface

6.2.1 Building the forms

(SQL*Forms has special terminology which is used here. Please refer to appendix F)

SQL*Forms represented the interface between the database and the users of those data. The design of the forms followed closely what had been described in the data flow diagrams. Seven forms with a total of twenty four pages were designed to meet the needs of this system.

The functions in each form is described briefly below. (Further details can be found in appendix J.)

MASTER form

The weekly production scheduling is done in this form. Page 1 presents the MPS. Values in the P and C fields are the planned and committed quantities of units respectively (figure 16). The ACT field displays the remaining quantities of units that have to be produced to match the planned quantities in the present week. At the end of the week, a negative value indicates that the production for that unit has fallen below the planned figure. While a positive value indicates over production.

With demand predictions on hand, user can update the MPS in page 1 (figure 16). User then move on to pages 2 and 3 to check whether the schedule places a realistic demand on the factory processes and ordering of materials. Negative values in the AVAIL fields in page 2 indicate that the processing time demanded by the schedule exceeds the maximum capacity allowable (figure 17). A positive value shows the available capacity in that process.

In page 3 user can scroll down the rows to see whether or not there is any material that has an order release in the WEEK0 field (figure 18). This indicates that the material should have been ordered earlier than the present week in order to support the current schedule.

User can amend the MPS in page 1 to make it more realistic. Exit from the form can be executed in page 3 (figure 18). Page 4 is created to enable user in the CAPACITY form to access a trigger in this form which compute the order releases of materials (figure 19).

MASTER PRODUCTION SCHEDULE

		28-MAY-90_				04-JUN-90_				11-JUN-90_				18-JUN-90_			
		OVERSEA		LOCAL		OVERSEA		LOCAL		OVERSEA		LOCAL		OVERSEA		LOCAL	
UNITNO	ACT	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C
9500__	-90	50_	0__	40_	20_	55_	6__	40_	8__	50_	0__	15_	0__	55_	0__	15_	0__
9505__	-90	50_	0__	40_	15_	55_	0__	40_	0__	50_	0__	15_	0__	55_	0__	15_	0__
9520__	-90	50_	24_	40_	0__	55_	6__	40_	0__	50_	0__	15_	0__	55_	0__	15_	0__
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SCHEDULE NEW WEEK ___

Figure 16 Page 1 of Master form

CHECK CAPACITY AND MATERIAL REQUIREMENT Y__

	28-MAY-1990	04-JUN-1990	11-JUN-1990	18-JUN-1990
	AVAIL	AVAIL	AVAIL	AVAIL
PACK	3.125	3.021	3.646	3.542
MACHINE#1	2.583	2.361	3.194	3.056
MACHINE#2	3.229	3.021	3.646	3.542

Figure 17 Page 2 of Master form

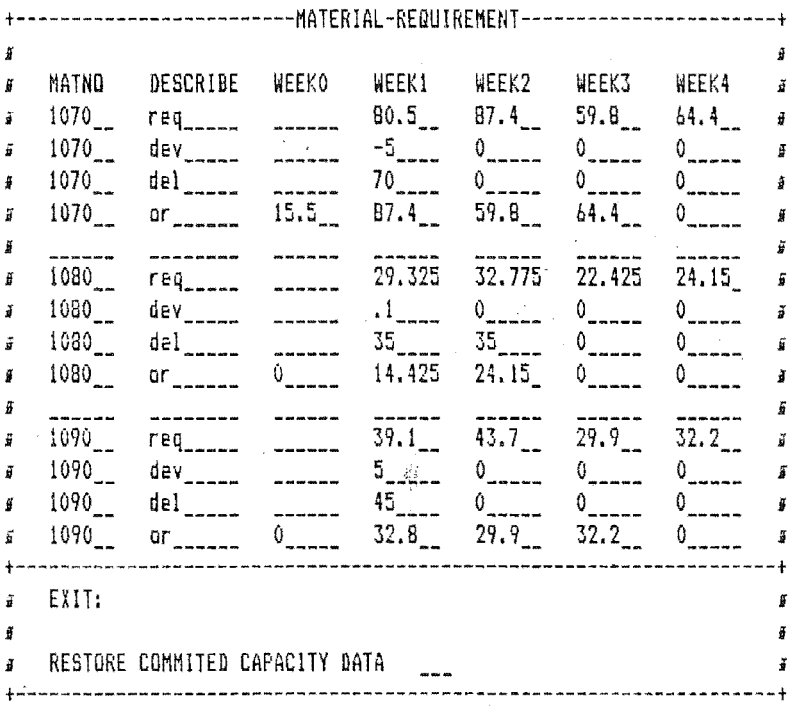


Figure 18 Page 3 of Master form

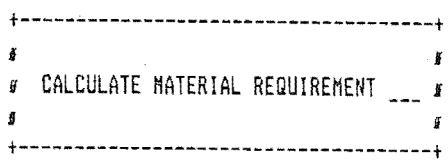


Figure 19 Page 4 of Master form

CREORDER form

The CREORDER and CAPACITY forms are designed for order processing. Page 1 of CREORDER form is for entering order's data (figure 20). A trigger at the PRODNO field checks that the product number is a valid one. A message will appear at the bottom of the page for an invalid product number.

On entering page 2 (figure 21), a trigger will process the order's data and display the computed values. Any changes to the orders can be made in page 3 (figure 22).

```
+--------------------ORDER'S-PARTICULARS-------------------------+
#                                                               #
#          L/O   1____      REQDATE   07-JUN-90                 #
#                                                               #
#         ORDERNO 109__      CUSTOMER farmer's__                #
#                                                               #
+---------------------------------------------------------------+

                INPUT PRODUCT AND QUANTITY
+---------------------------------------------------------------+
#   ORDERNO  PRODNO   NAME                          PQTY        #
#   109____  19533__  LUNCH BOX S10*2,S11*3_____   10________  #
#   109____  9525___  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
#   -------  -------  ------------------------    ----------  #
+---------------------------------------------------------------+

no such product! try again_____________________________________________
```

Figure 20 Page 1 of Creorder form

CHECKING ORDER'S

		NO.	VOL. AVAIL.
VOLUME 140___ cu.m.	CONTAINER TYPE A (50 cu.m.)	3__	10_____ cu.m.
	CONTAINER TYPE B (100 cu.m.)	2__	60_____ cu.m.
WEIGHT 180___ kg.			

COSTING

MATERIAL COST	169.2___
MOULDING COST	21.66___
PACKAGING LABOUR COST	9.17____
OVERHEAD COST	27_____
TOTAL COST	227.03__

Figure 21 Page 2 of Creorder form

CHANGE

ORDERNO	PRODNO	PQTY
109_____	19533__	10_____
109_____	19525__	15_____
109_____	19500__	10_____
----------	-------	----------
----------	-------	----------
----------	-------	----------
----------	-------	----------
----------	-------	----------
----------	-------	----------
----------	-------	----------

Figure 22 Page 3 of Creorder form

CAPACITY form

Order processing continues in the CAPACITY form. It checks whether or not factory capacity and material stock can support an order. Top of page 1 presents the available packaging and machine capacities (figure 23). By entering 'y' to the CALC field, the required capacity of an order is calculated. User then loads the order into the production schedule week(s) where there is enough capacity to support it. Of course, the chosen week(s) will need to be before the delivery date.

Function keys have been redefined such that user cannot access pages 2 and 3. The purpose of these two pages is for computations (figures 24 and 25).

Any future shortage of material or capacity as a result of loading an order will be shown in page 4 (figure 26). The user can amend the figures in the master schedule, in page 5, to try to rectify the shortages (figure 27). At the top of page 5, estimation of the packaging starting and completion dates for the order is also displayed.

If it is found that the material and various processes can support the manufacturing of an order satisfactorily, the user can either commit the order (option 1 in figure 26) or leave without committing any

order (option 2). The User leaves the form from page 6 (figure 28).

```
+-------------------+-----AVAILABLE---CAPACITY-------+----------+
#                   #         #         #         #          #
#                   # 28-MAY-90# 04-JUN-90# 11-JUN-90# 18-JUN-90 #
#                   # WEEK1    # WEEK2    # WEEK3    # WEEK4    #
#PACKAGING (LOCAL)  # .618__   # .789__   # .312__   # .312__   #
#PACKAGING (OVERSEAS)# 3.967_  # 4.083_   # 4.688_   # 4.688_   #
#MACHINE # 1        # 4.581_   # 4.766_   # 5_____   # 5_____   #
#MACHINE # 2        # 4.666_   # 4.917_   # 5_____   # 5_____   #
#                   # ______   # ______   # ______   # ______   #
+-------------------+---------+---------+---------+----------+
+-------------------------REQUIRED-CAPACITY----------------------+
#                                                    DELIVERY    #
#ORDER NO109___        PACKAGING    .917__          07-JUN-1990__ #
#                      MACHINE # 1  1.333_                       #
#L/O    1__            MACHINE # 2  .833__                       #
#                                   ------                       #
#CALC ?  y__                        ------                       #
+-----------------------------------------------------------------+
+-----------------------------------------------------------------+
#            CHOOSE:   WEEK # 1__  AND  2__                       #
#                                                                 #
#                      LOAD   y__                                 #
+-----------------------------------------------------------------+
```

Figure 23 Page 1 of Capacity form

```
                 ======== CAPACITY1 ========

 WEEK1    WEEK2    WEEK3    WEEK4    CALC1    CALC2    REQ
 .618__   .789__   .312__   .312__   .618__   .789__   .917___
 3.967_   4.083_   4.688_   4.688_   3.967_   4.083_   .917___
 4.581_   4.766_   5_____   5_____   4.581_   4.766_   1.333__
 4.666_   4.917_   5_____   5_____   4.666_   4.917_   .833___
 ------   ------   ------   ------   ------   ------   -------

                 ========  WEEK  ========

   WEEK1 28-MAY-90                    WEEK2 04-JUN-90

   WEEK3 11-JUN-90                    WEEK4 18-JUN-90

   CALC1 28-MAY-90                    CALC2 04-JUN-90
```

Figure 24 Page 2 of Capacity form

```
                    ======= MPS ========

4 0 4 2 5 6 4 8 5 0 1 0 5 0 1 0 4 0 4 2 5 6 4 8 6
4 0 4 1 5 0 4 0 5 0 1 0 5 0 1 0 4 0 4 1 5 0 4 0 3
4 2 4 0 5 6 4 0 5 0 1 0 5 0 1 0 4 2 4 0 5 6 4 0 4
- - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - -

                    ======== MRP ========

 WEEK1    WEEK2    WEEK3    WEEK4    WK1      WK2      CALC1 CALC2
 78.2__   87.4__   59.8__   64.4__   78.2___  87.4__   6.9   0__
 22.425_  32.775   22.425   24.15_   22.425_  32.775   -3.   0__
 36.513   43.7__   29.9__   32.2__   36.513_  43.7__   2.0   0__
 34.76_   13.13_   3.25__   3.5___   34.76__  13.13_   23.   0__
 15.8__   1.9___   1.3___   1.4___   15.8___  1.9___   9.5   0__
 17____   4.05__   3.25__   3.5___   17_____  4.05__   9__   0__
 245___   285___   195___   210___   245____  285___   5__   0__
 100___   95____   65____   70____   100____  95____   20_   0__
 75____   95____   65____   70____   75_____  95____   -5_   0__
 ------   ------   ------   ------   -------  ------   ---   ---
```

Figure 25 Page 3 of Capacity form

```
  CHECK MATERIAL  y__                ___  (press enter to
                                           go to master form)
+--------------MATERIAL-SHORTAGE---------------------+
#                                                    #
#              MATERIAL   QUANTITY                   #
#              1070____   13.2____                   #
#              --------   --------                   #
#              --------   --------                   #
#              --------   --------                   #
#              --------   --------                   #
#              --------   --------                   #
#              --------   --------                   #
#              --------   --------                   #
+----------------------------------------------------+
+----------AVAILABLE-CAPACITY-AFTER-LOADING----------+
#                                                    #
#          PROCESS        1ST WK   2ND WK            #
#          PACK (LOCAL)   0_____   .483__            #
#          PACK (O.S. )   3.974_   4.083_            #
#          MACHINE #1     3.692_   4.322_            #
#          MACHINE #2     4.11__   4.64__            #
+----------------------------------------------------+
```

Figure 26 Page 4 of Capacity form

```
+---------------------------------PRODUCTION-------------------------------+
|                START              FINISH            DELIVERY            |
|                28-MAY-90          04-JUN-90         07-JUN-90           |
+-------------------------+-------------------------------+-------------+
                          |     UPDATE MPS ____           |
                          |                               |
                          |                               |
+--------+----------------+---------------MASTER-SCHEDULE-+-------------+
|        |                |                               |             |
|        |  ORDER'S QTY   |       28-MAY-1990             |   04-JUN-1990     |
|        +--------+-------+                               |             |
|        | FIRST  | SECOND |  OVERSEAS    LOCAL            | OVERSEAS    LOCAL     |
| UNITNO | WEEK   | WEEK   |  PLN  COMM PLN  COMM          | PLN  COMM PLN  COMM   |
| 9500__ | 40__   | 20__   |  40__ 0___ 40__ 20__          | 55__ 6___ 40__ 8___   |
| 9505__ | 20__   | 10__   |  40__ 0___ 40__ 15__          | 55__ 0___ 40__ 0___   |
| 9520__ | 30__   | 15__   |  40__ 24__ 40__ 0___          | 55__ 6___ 40__ 0___   |
| ______ | ____   | ____   |  ____ ____ ____ ____          | ____ ____ ____ ____   |
| ______ | ____   | ____   |  ____ ____ ____ ____          | ____ ____ ____ ____   |
|        |        |        |                               |                       |
+--------+--------+--------+-------------------------------+-----------------------+
|        RECALCULATE CAPACITY AND MATERIAL REQ. ___                               |
+---------------------------------------------------------------------------------+
```

Figure 27 Page 5 of Capacity form

```
LEAVING FORM

1. COMMIT ORDER INTO MASTER SCHEDULE  y__

2. DO NOT COMMIT ORDER                ___

                                      ___ (recalc mrp)
```

Figure 28 Page 6 of Capacity form

FACTORY form

Pages 1 and 2 are used to enter production data at the end of every shift (figures 29 and 30). This updates the stock levels and the relevant tables.

Pages 3 and 4 allow the user to query the system as to whether or not any part or material stock is outside their maximum or minimum levels (figures 31 and 32).

Page 5 shows the work-in-process. The PACKED field shows the ratio of orders that has been packed. A PACKED value of one means that the order is fully packed and is waiting to be delivered.

MOULDING MACHINES DATA INPUT

PROCEED? y__ MACHINE NO. 1___

MACHINE COUNTER

PARTNO	START	FINISH	REJECTS	TOTAL
500____	1100___	1120___	2_____	18_____
501____	1450___	1480___	3_____	27_____

UPDATE STOCK ___

Figure 29 Page 1 of Factory form

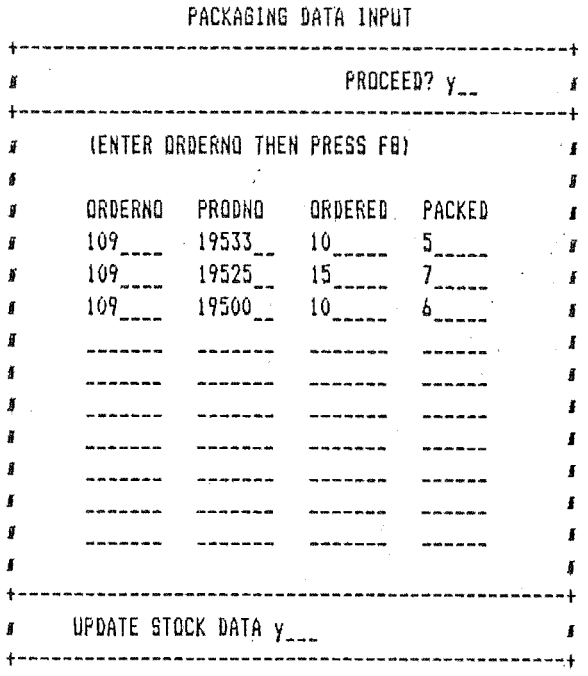


Figure 30 Page 2 of Factory form

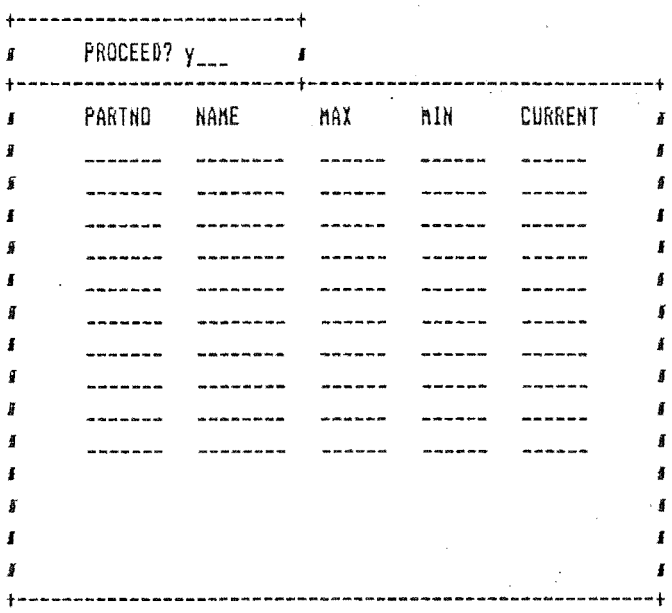


Figure 31 Page 3 of Factory form

MATERIAL OUTSIDE STOCK LEVELS

PROCEED? y___

MATNO	NAME	MAX	MIN	CURRENT
1040__	box-35__	125___	75____	22_____
1060__	poly-bag	125___	75____	70_____
------	--------	------	------	-------
------	--------	------	------	-------
------	--------	------	------	-------
------	--------	------	------	-------
------	--------	------	------	-------
------	--------	------	------	-------

Figure 32 Page 4 of Factory form

ORDERS BEING PACKED OR WAITING TO BE DELIVERED

ORDERNO	CUSTOMER	DELIVERY	PACKED
109____	farmer's__	07-JUN-90	.498____
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------
-------	----------	---------	--------

Figure 33 Page 5 of Factory form

REORDER form

At the beginning of each week, this form is used to order materials. Querying the top block of page 1 brings up materials that have order releases in the present week. At the bottom of the page, the user can modify the quantities to order or nominate a different supplier (figure 34).

Page 2 is the material order form where the user can call up the purchases for the week and get them printed (figure 35).

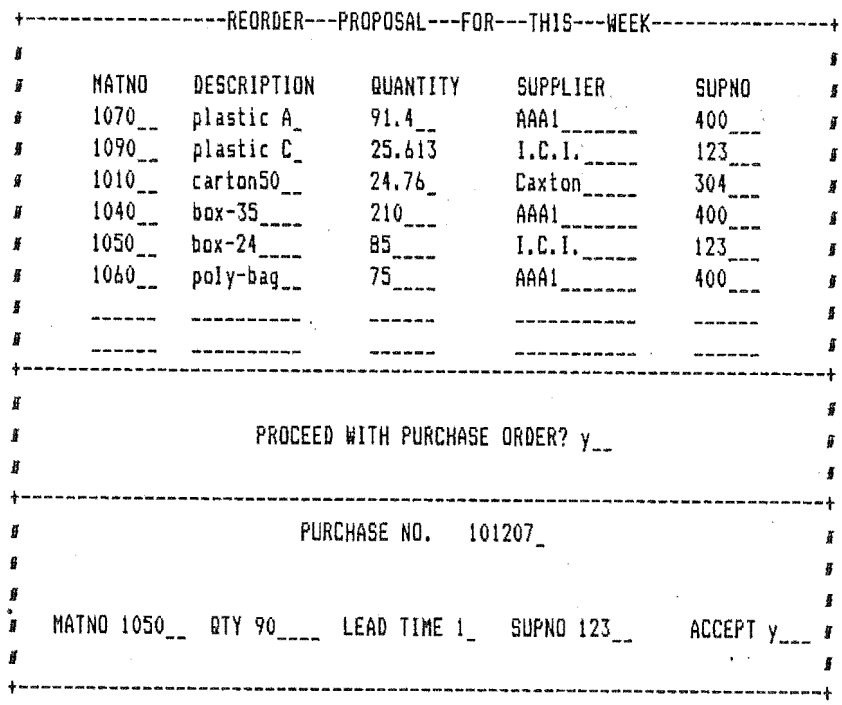


Figure 34 Page 1 of Reorder form

```
+--------------------XXXXX-MOULDING-LTD.,---------------------+
#                       CHRISTCHURCH                           #
+--------------------------------------------------------------+
#  SUPPLIER                                                    #
#  Name   : I.C.I._____                                       #
#  Address: Christchurch_____          Date: 28-MAY-90__      #
#                                                              #
#                                                              #
#  Purchase no: 101207_______                                  #
+--------------------------------------------------------------+
#  Matno    Name       Quantity  Unit cost   Cost             #
#  1050___  box-24___  90_____   .01_______  .9_______        #
#  1090___  plastic C  30_____   2________   60_______        #
#  _______  ________   _______   _________   _________        #
#  _______  ________   _______   _________   _________        #
#  _______  ________   _______   _________   _________        #
#                                                              #
#                                  Total :  60.9_____          #
#                                                              #
+--------------------------------------------------------------+
```

Figure 35 Page 2 of Reorder form

CONTROL form

This form is for entry of data from goods coming in or going out of the factory. Page 1 updates the stock levels when quantities of received material are entered (figure 36). Page 2 records orders that have been delivered (figure 37).

TODAY'S DATE 28-MAY-1990

```
+--------------INWARD-GOODS-----------------------+
#                                                 #
#                                                 #
#       PURCHASE NO. 101207____                   #
#                                                 #
#       PURNO      MATNO    QTY      RECEIV       #
#       101207__   1050__   90____   90____       #
#       101207__   1090__   30____   30____       #
#       --------   ------   ------   ------       #
#       --------   ------   ------   ------       #
#       --------   ------   ------   ------       #
#                                                 #
#                                                 #
#                                                 #
#                                                 #
+-------------------------------------------------+
```

Figure 36 Page 1 of Control form

ORDERS THAT HAVE BEEN DELIVERED

```
+-------------------------------------------------+
#      (F8 then down key to retrieve record)      #
#                                                 #
#      ORDER NUMBER 107__         DELIVERED y__   #
#                                                 #
#                                                 #
+-------------------------------------------------+
```

Figure 37 Page 2 of Control form

PERFORMANCE form

Page 2 presents the performance indices for each week (figure 39). All the indices, except the material wastage index, are automatically computed when the MPS is rescheduled in page 1 of Master form (figure 16). The material wastage index is calculated in page 1 of performance form (figure 38).

```
+----------------------MATERIAL--WASTAGE----------------------+
#  PROCEED? y__                                               #
+-------------------------------------------------------------+
#  MATNO    NAME       USE         PRODUCE      WASTAGE       #
#  1070__   plastic A  8.1_____   7.2______    .111111       #
#  1080__   plastic B  7.3_____   6.9______    .054794       #
#  1090__   plastic C  8.9_____   8.3______    .067415       #
#  ------   ---------  ---------  ----------   -------       #
#  ------   ---------  ---------  ----------   -------       #
#  ------   ---------  ---------  ----------   -------       #
#  ------   ---------  ---------  ----------   -------       #
#  ------   ---------  ---------  ----------   -------       #
+-------------------------------------------------------------+
#    UPDATE PERFORMANCE DATA y__                              #
+-------------------------------------------------------------+
```

Figure 38 Page 1 of Performance form

```
+----------------------FACTORY---PERFORMANCE---------------------------+
#                                                                      #
#                                                                      #
#             PACKAGING   MOULDING    MATERIAL     UTILIZATION         #
#  WEEK       (ACT/PLN)   (ACT/PLN)   WASTAGE    MACHINE#1   MACHINE#2 #
#  14-MAY-90  .812___     .854___     .025___     .852___     .867___   #
#  21-MAY-90  1.02___     .972___     .015___     .885___     .785___   #
#  28-MAY-90  .315___     .237___     .078___     .07____     .087___   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#  ---------  -------     -------     -------     -------     -------   #
#                                                                      #
#                                                                      #
#                                                                      #
#                                                                      #
+----------------------------------------------------------------------+
```

Figure 39 Page 2 of Performance form

6.2.2 Building the menu

The menu created has two levels. The top level is the application menu and the bottom the main menu.

APPLICATION MENU

A. Factory Data
B. Master Production Schedule
C. Create New Order
D. Factory Performance

MAIN MENU

A. 1.	input of production data	(FACTORY)
2.	input of inward/outward goods data	(CONTROL)
3.	material purchase proposals	(REORDER)
B. 1.	master production schedule	(MASTER)
C 1.	create new order	(CREORDER)
2.	load new order	(CAPACITY)
D 1.	factory performance	(PERFORMANCE)

In brackets are the forms that can be accessed through the main menu.

Building the menu was easy. The applications were created first as shown in figure 40. Then the various work-classes and the users in each work-class who can access the applications were specified (figure 41). DBM, OSC and BGM were various privileges that a user can have. They are explained in appendix I.

```
SQL*MENU - APPLICATION SPECIFICATION


          Application: NEWORD
        Creation_date: 04-APR-90
              Creator: SYSTEM_________
    Version.release_nr: 1.1__
     Last_release_date: _________

Menu-directory: ________________________________________

Identification: Create New Order_____________________
```

Figure 40 SQL*menu-application specification

```
SQL*MENU - WORK CLASS/USER INFORMATION        07-JUN-90

 Application_name: NEWORD

Work_class : 10__
Description: non-management employee_____________________________
             ------------------------------------------------------
             ------------------------------------------------------
             ------------------------------------------------------

 Work_class  User_name            DBG OSC BGM
 10__        CLERK_____________   N   N   N
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
 ----        --------------------  -   -   -
```

Figure 41 SQL*menu-work class/user information

```
                Options of application: NEWORD    menu: NEWORD

Option      : 1_ Lower work-class: 10_ Higher work-class: 20_ Cmd_type: 4
Option_text : Create New Orders_____________________________________________
Command_line: runform -m creorder &un/&pw___________________________________

Option      : 2_ Lower work-class: 10_ Higher work-class: 20_ Cmd_type: 4
Option_text : Load New Orders_______________________________________________
Command_line: runform -m capacity &un/&pw___________________________________

Option      : ___ Lower work-class: ____ Higher work-class: ____ Cmd_type: _
Option_text : _____________________________________________________________
Command_line: _____________________________________________________________

Option      : ___ Lower work-class: ____ Higher work-class: ____ Cmd_type: _
Option_text : _____________________________________________________________
Command_line: _____________________________________________________________

General menu info <<                                    >> options help
```

Figure 42 SQL*menu-options for application

Finally, the various forms were attached to the applications (figure 42). The LOWER and HIGHER work-classes specified the range of work-class that can access the forms.

All the information that is needed to create this menu is given in the documentation generated by SQL*menu, in appendix I.

CHAPTER VII

EVALUATION

7.1 System Capability

Test data that was put through the system and the results generated (appendix J) show that the system designed in this project has been able

1) to meet the computational requirements of an MRP system, and
2) to provide solutions to the two functional areas in the business: order processing and production planning and control.

It is acknowledged that to fully meet the business requirements, a trial period is needed where real data can be used and the applications fine-tuned. However, this was not possible as the business concerned was under receivership and was therefore unable to provide the required assistance.

7.2 Approaches to the System Development

The basic objective of the system and structured design approaches was to develop an integrated system that would meet the information needs of the business. (A system is integrated if the integration of its

subsystems took advantage of the interrelatedness and interdependence between the subsystems.) At various stages of the system development, the two approaches provided techniques and a sense of direction towards this objective.

The system approach to the business analysis has led to a detail understanding of the organisation. It revealed the information deficiencies, the business constraints and the interrelationship between the personnels in the organisation.

This detailed understanding enabled an information system to be designed and built to meet all the information needs of the business. The top-down approach to the system designing and construction required the higher level processes to be integrated before moving on to the lower level processes. The normalisation process ensured that the tables in the database were well integrated. This led to the development of an integrated system.

7.3 Diagramming techniques

The three types of diagramming worked well with the relational database approach and the Oracle software tools. The entity-relationship and data flow diagrams provided the entities/files and data types, which were essentially the tables and columns of the database. The

decomposition and data flow diagrams provided a graphic specification for building the forms and the menu.

In most design work, it is unlikely to have the right design after the first trial. The finalised diagrams, tables and forms came after a few iterations of the development process. The diagrams proved to be useful during this period. The diagramming techniques were user-friendly and it was easy to make changes to the diagrams. They provided a useful ground for trying out new ideas.

7.4 Oracle Software Package

7.4.1 Oracle flexibility

Oracle flexibility has allowed changes to be made easily. The extensive SQL and SQL*Plus commands enable the database to be restructured with ease. The MODIFY mode in SQL*Forms provided useful functions such as adding or deleting fields and the ability to cut and paste any part of the form. These functions were used extensively.

This flexibility should allow the use of Oracle to expand along with the business. Other functional areas such as finance, sales and business planning can be integrated into the present system when the needs arise. For example, if the company wants to keep track of the

sales made by each salesperson, this can be done by creating:

1) a new SALES table with columns of SALESPERSON_ID, NAME and SALES_VOLUME,

2) SALESPERSON_ID and SALES_VOLUME fields in the CREORDER form for entering the data, and

3) a trigger to update the table whenever there is a new order.

7.4.2 Ease of Use and Learning

One drawback that was encountered in Oracle was that it has many types of trigger that can be defined at three different levels, i.e. form, block and field levels. Also certain commands could not be used in certain types of trigger. Experimenting with the multitude of trigger types had prolonged the learning period.

A comparison of different 4GL softwares shows that in the report generating area, Oracle is more difficult to use and learn (see appendix D). The test report that was used in that comparison was complicated. In this project the report requirements were quite straight forward. It was found to be fairly easy to use and to create these reports. The commands that were used in these reports are shown in appendix J.

In general, it could be said that the high degree of sophistication offered by Oracle, which was only partly needed in this project, has lengthened the development period due to the long learning process.

7.4.3 Processing Speed

The processing speed in this system was satisfactory. Triggers with the highest number of commands like those in the MASTER form took about 20 seconds to compute the capacities and material requirements. This should meet the requirements of the business.

7.5 End-user System Development

This project has offered a method and techniques in developing an MRP system. Small manufacturing companies can use this method and techniques to design and build their own systems. However, they will need suitable software packages as platforms to build their systems.

Oracle appears to have the necessary functions and the flexibility to meet the information needs of these companies as demonstrated by this project. Because of its sophistication, it may require system developers who are familiar with computing.

The ideal software package would be one that is simple enough for nontechnical end-users to develop the system, flexible and powerful enough to meet the needs of the business and not too costly. This may not be too difficult to achieve as software packages are continually being developed towards this ideal. Already available are so-called flat-file databases [13]. They allow end-user development of simple information system. They are less flexible but easier to learn and are cheaper than relational databases.

The recommendation from this project is for more research and assessment of 4GL tools that can be used for end-user development of MRP systems. This will lower development cost by not having to bring in system developers from outside the company.

7.6 Conclusion

Having the most suitable software tools will not guarantee a successful information system. Developers need to have the right approach and techniques. This was the main concern of this project.

In conclusion it can be said that by using the system and structured design approaches, an integrated system was developed to meet the information needs of the business studied.

The three types of diagrams, entity-relationship, decomposition and data flow diagrams, provided sufficient information to build the relational database and the forms.

The Oracle software package was found to be adequate as a platform to build the system. However, it was felt that the software tools were too sophisticated for the requirements of this project.

Summarised below are the approaches and techniques that have been used.

Development philosophy:

System approach

Structured design

Graphic tools:

Information and material flow diagram

Building plan

Organisational chart

Decomposition diagram

Entity-relationship diagram

Data flow diagram

Production planning and control mechanism:

MRP type II (closed loop system)

Data structure:

Relational database approach

Software package:

Oracle Database management system

(Fourth generation language)

REFERENCES

1. Scott Hamilton, Computerworld Special Report: Manufacturing (Computerworld New Zealand, No. 71, May 30 1988).

2. Meilir Page-Jones, The Practical Guide to Structured Systems Design (Yourdon Press, 1980), Chapter 1.

3. Edward J. Hay, The Just-In-Time Breakthrough (John Wiley & Sons, 1988), Chapter 9.

4. Roger G. Schroeder, Operation Management: Decision Making in the Operation Function, 1989, Chapter 15.

5. A. K. Kochhar, Development of Computer-Based Production Systems (Edward Arnold, 1979), Chapter 5.

6. James T. Perry, Understanding Oracle (Tech Publication, 1989 ed.), Chapter 2.

7. Daniel J. Cronin, Mastering ORACLE (Hayden Books, 1989), Chapter 9.

8. D. R. Howe, Data Analysis for Data Base Design (Edward Arnold, 1983), pp 6 - 16.

9. D. Mason, INFO 312 Lecture Handouts (Victoria University, 1989).

10. Computer-Aided Software Engineering Symposium, Wellington, New Zealand (Digital Consulting, Inc., 1987).

11. S. K. Misra, Increasing the Power of Fourth-Generation Report Generators (Information and Software Technology, Vol. 31 No.4, 1989).

12. ORACLE for Personal Computers Promotional Brochures (Oracle Corp., 1989).

13. S. Palmer, New Depth to Flat-File Databases (PC World New Zealand, No. 9, June 1989).

14. James Martin, Diagramming Techniques for Analysts and Programmers (Prentice-Hall, 1985), Chapter 20.

15. Daniel J. Cronin, Mastering ORACLE (Hayden Books, 1989), Chapter 7.

BIBLIOGRAPHY

1. Edward Yourdon, Managing the Structured Techniques (Yourdon Press, 1976).

2. J. Burch and F. Strater, Information Systems: Theory and Practice (Hamilton Publishing Co., 1974), Chapters 3 and 4.

3. C. J. Date, An Introduction to Database Systems (Addison-Wesley, 1981).

4. Joseph Orlicky, Material Requirements Planning (McGraw-Hill, 1975).

5. James H. Greene, Production & Inventory Control Handbook, 2nd Ed. (McGraw-Hill, 1987), Chapters 13 and 14.

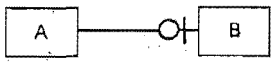
APPENDIX A

Entity-Relationship Diagram Keys [14]

Basic Associations

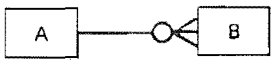
For each occurrence of A there is one and only one occurrence of B.

For each occurrence of A there is one or zero occurrence of B.



For each occurrence of A there is one or multiple occurrences of B.

For each occurrence of A there is zero, one, or multiple occurrences of B.



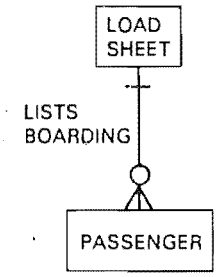
(Continued)

The relationship of A to B is uninteresting. (This is the tail end of a connector whose head end is necessary.)

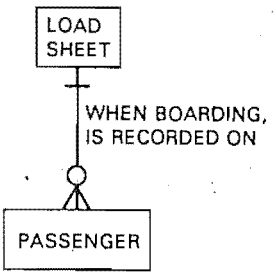
The relationship of A to B is of interest but is as yet unknown.

Labeled Associations

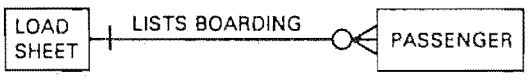
A label may be written on the left side of a link going *down:*



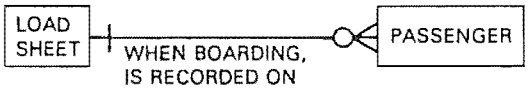
On the right side of a link going *up:*



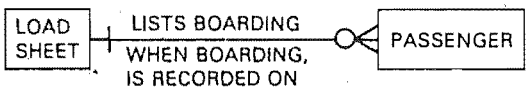
On the top of a link going left to right:



On the bottom of a link going right to left:



Two complementary labels may be written on the same link:



The labels should be composed to form a sentence.

APPENDIX B

Detail Data Flow Diagrams (DFD)

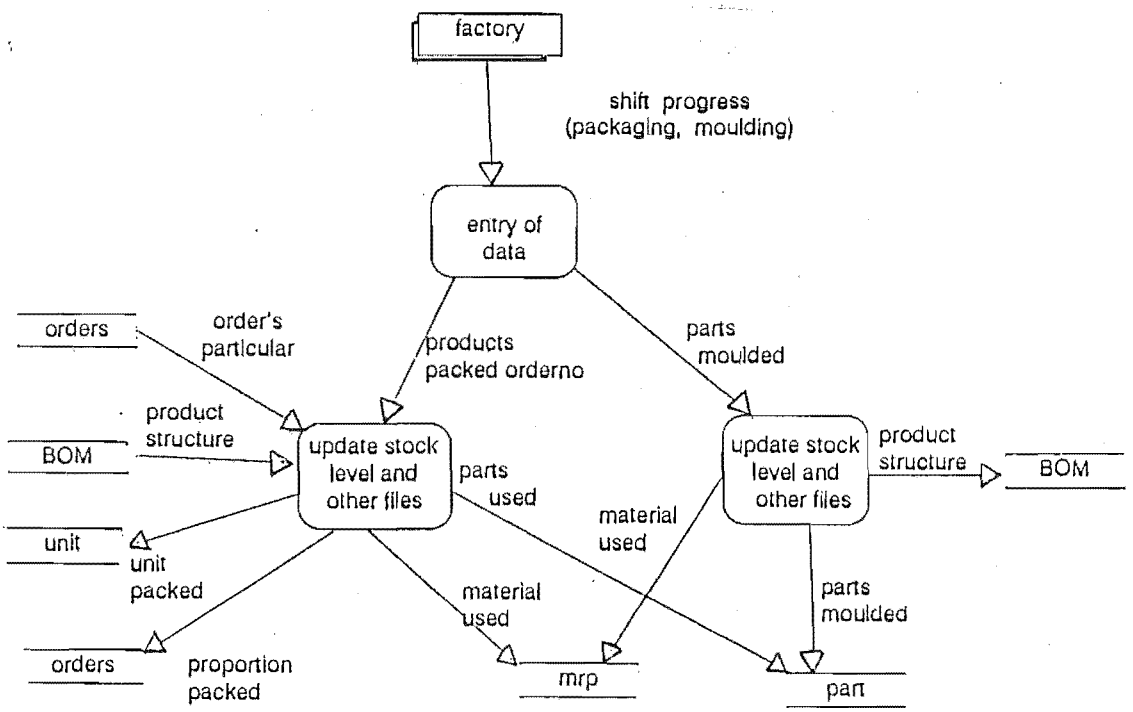
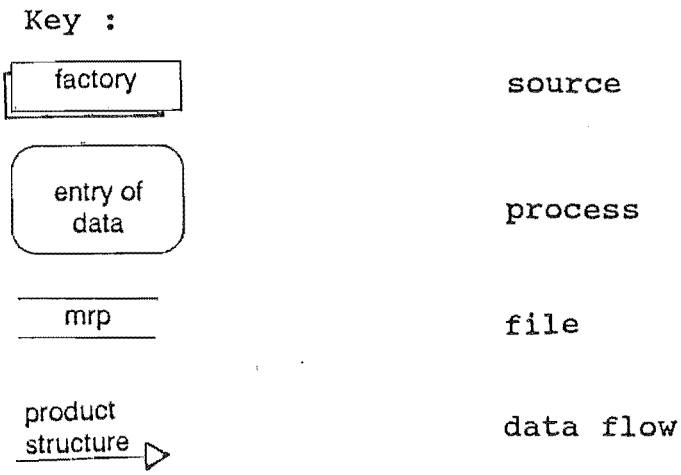


Figure B1 DFD of shift progress data entry



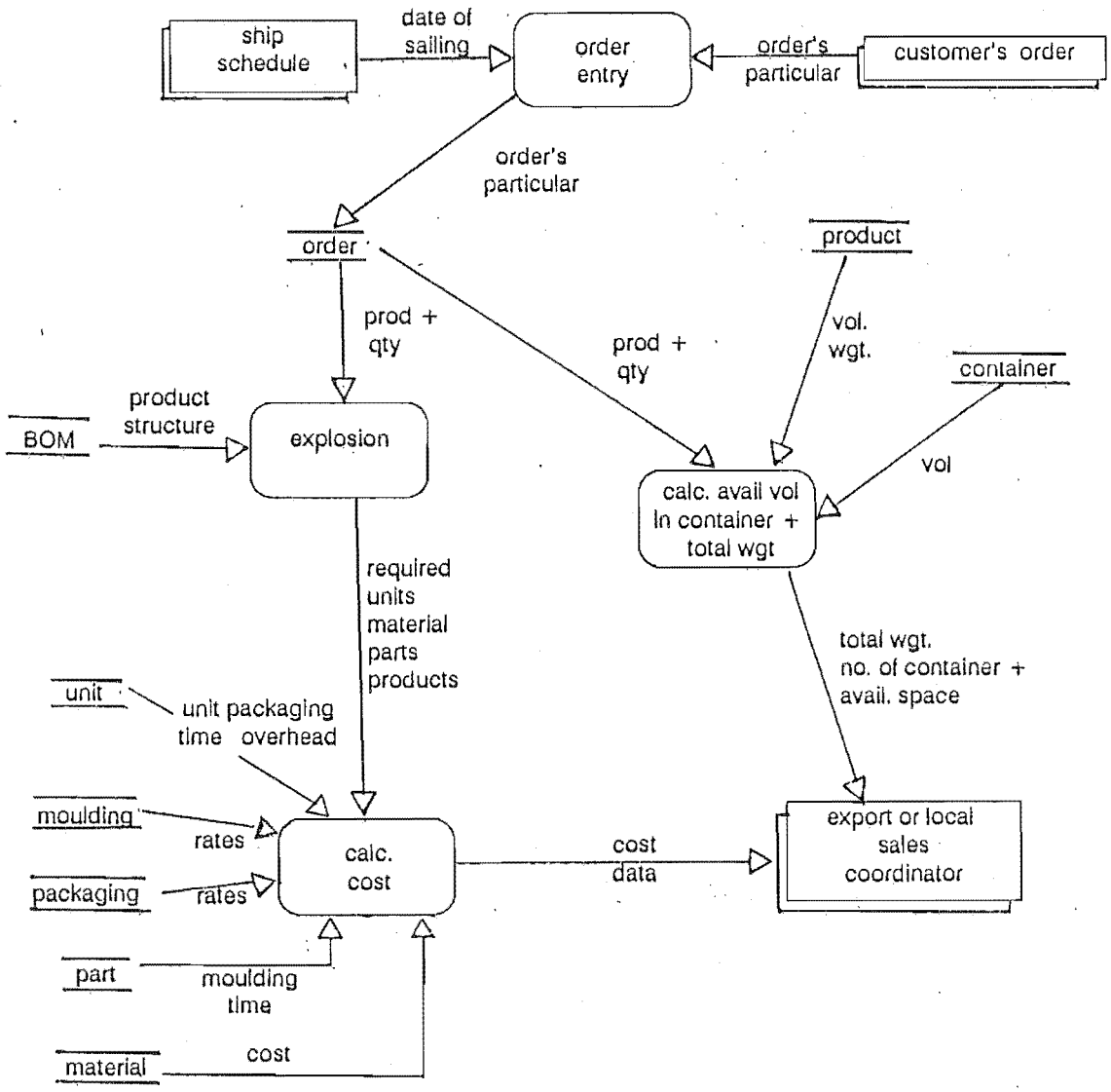


Figure B2 DFD of processing order

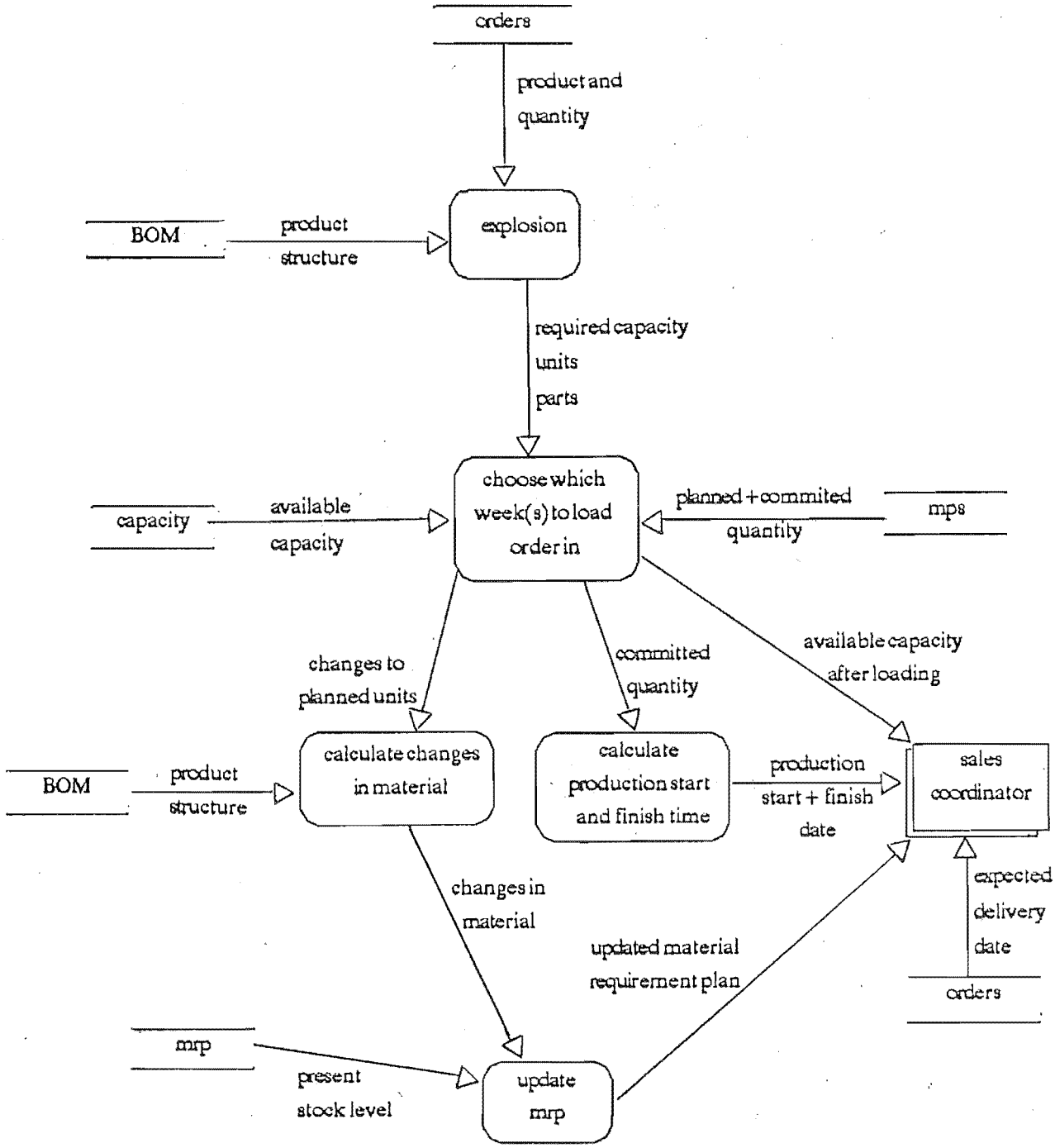


Figure B3 DFD of committing order

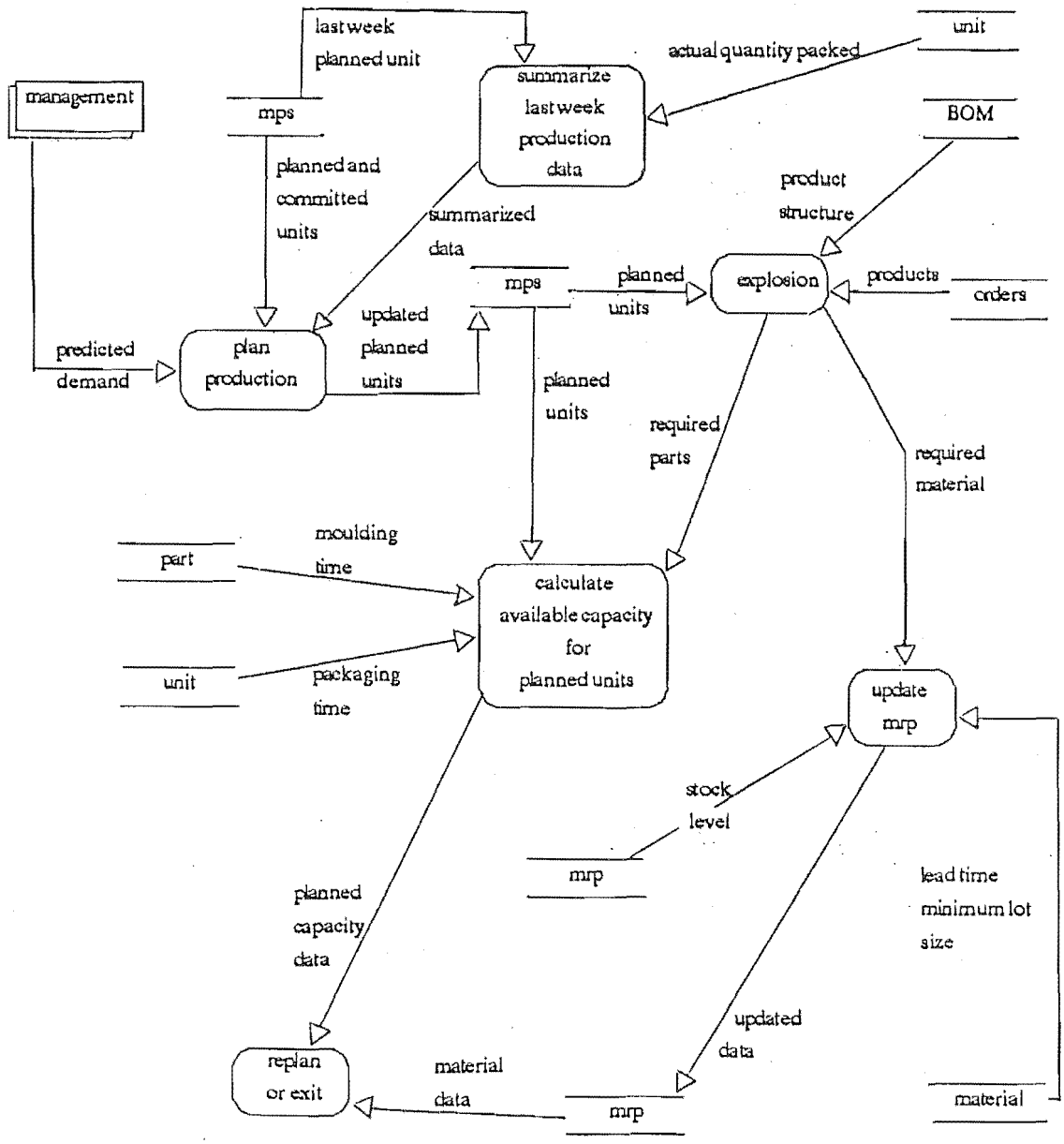


Figure B4 DFD of scheduling

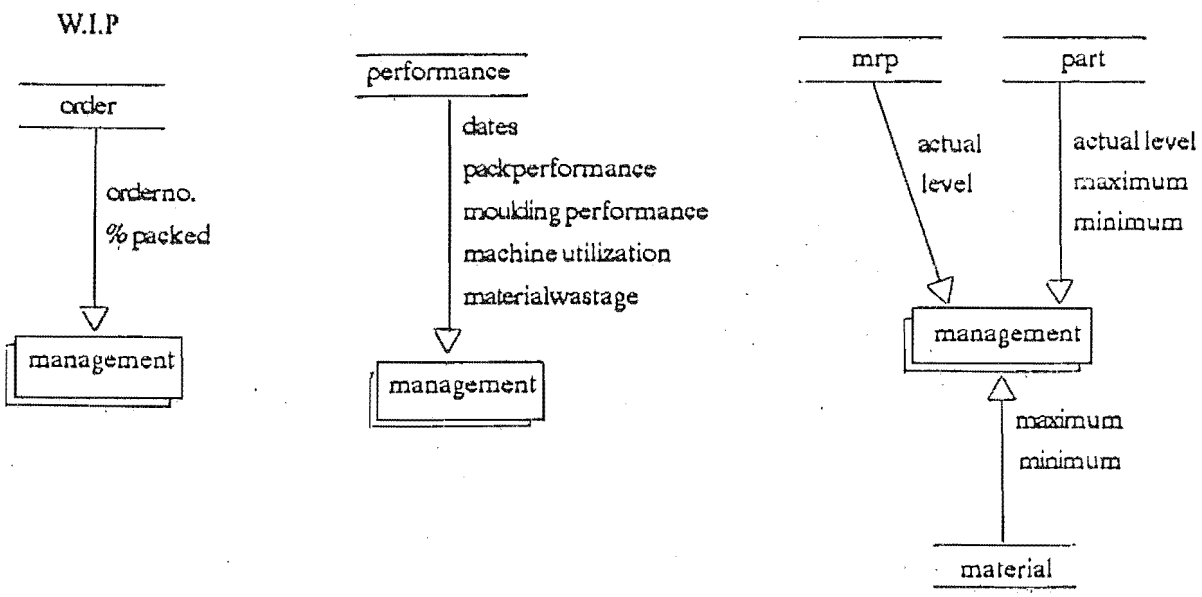


Figure B5 DFD of reports

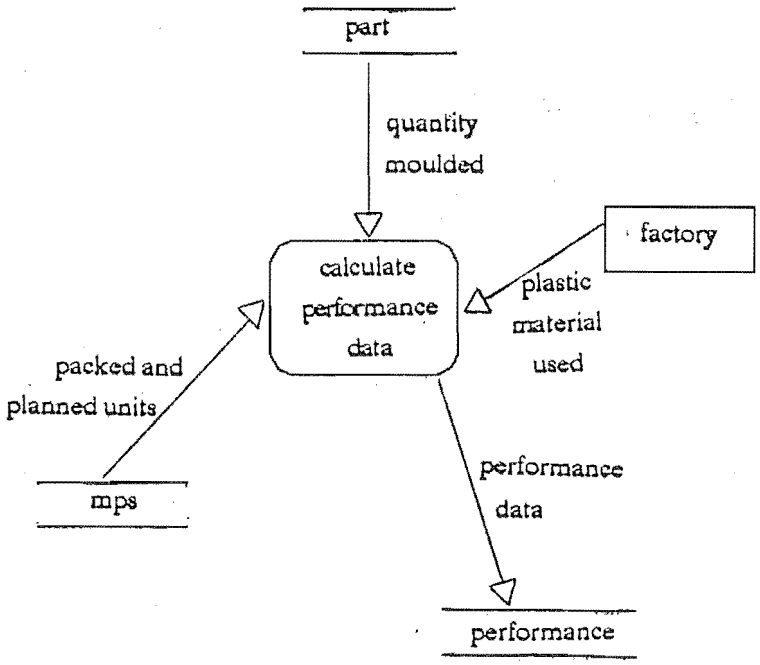


Figure B6 DFD of calculating performance data

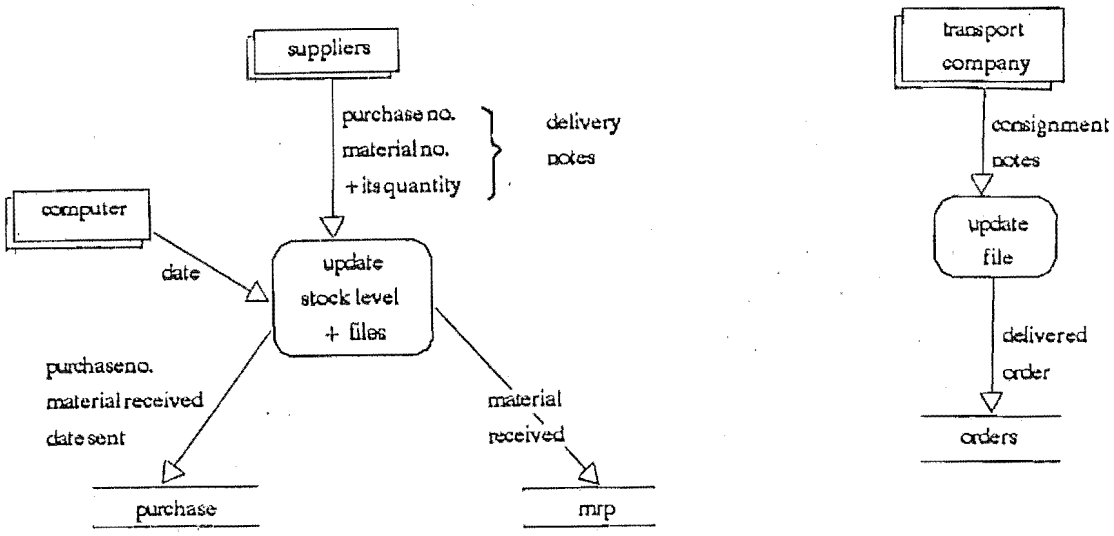


Figure B7 DFD of inwards/outward goods data entry

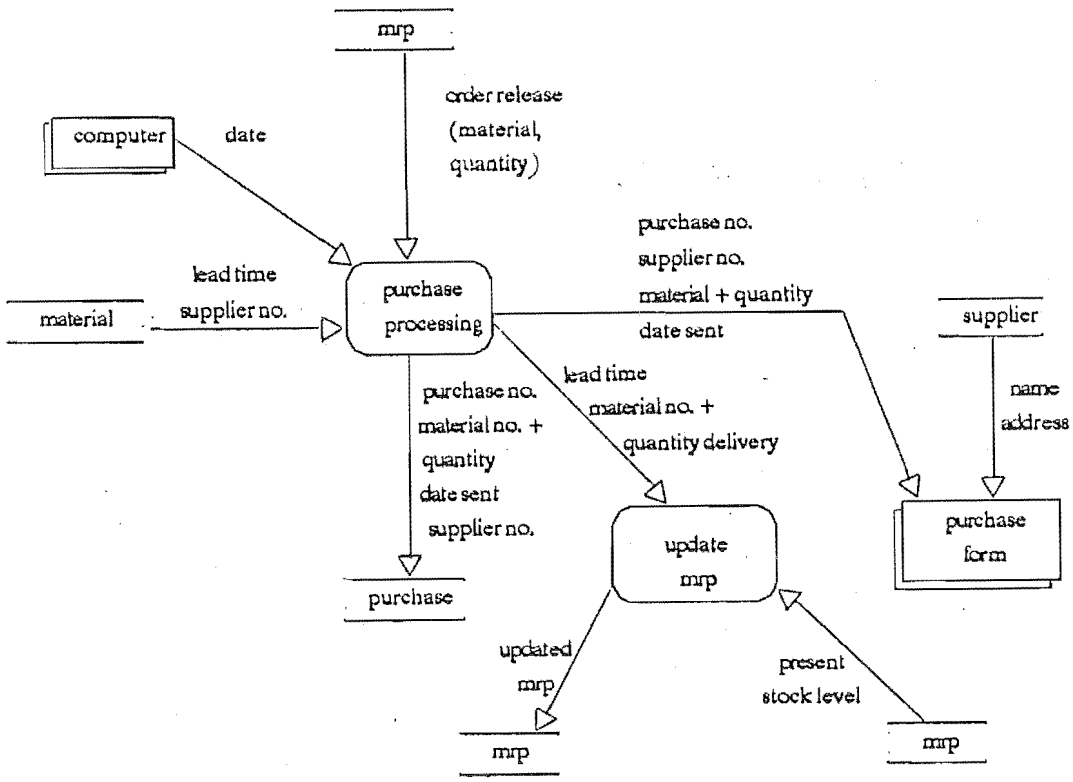


Figure B8 DFD of processing purchase form

APPENDIX C

Ship Schedules

PACIFIC MARITIME

CHRISTCHURCH OFFICE
TEL. (03) 797-071
(03) 650-642
FAX. (03) 650-641
TELEX 4615

172 HEREFORD ST CHRISTCHURCH

SCHEDULE UPDATE

TO: 10/05/89

ABC CONTAINERLINE

	DELORIS V220	TNT EXPRESS V221	ELLEN HODIG V222	HELEN V223	BRUSSEL V224	CORNELIS VAROLME V225
AUCKLAND	SLD	SLD	24/05	02/06	14/06	29/06
PT CHALMERS	SLD	11/05	27/05	05/06	17/06	03/07
NEW ORLEANS	-----	-----	-----	-----	-----	-----
GULFPORT	20/05	01/06	13/06	29/06	11/07	25/07
CHARLESTON	24/05	04/06	17/06	03/07	15/07	29/07
PHILADELPHIA	26/05	07/06	19/06	05/07	17/07	31/07
HALIFAX	28/05	09/06	22/06	07/07	20/07	03/08
FLUSHING	-----	17/06	-----	15/07	-----	11/08
ZEEBRUGGE	06/06	18/06	01/07	16/07	29/07	12/08
HAMBURG	10/06	21/06	05/07	19/07	01/08	15/08
FELIXSTOWE	09/06	23/06	07/07	21/07	03/08	17/08
FOS	17/06	29/06	13/07	27/07	09/08	23/08
LIVORNO	19/06	01/07	15/07	29/07	11/08	25/08

*** ELLEN HUDIG V222 HAS BEEN DELAYED IN AUSTRALIA ***

CHINA OCEAN SHIPPING COMPANY

	XFK V103	XSK V121	ZGJK V92	TPK V117	GBK V105	ZJK V132
LYTTELTON	SLD	SLD	24/05	09/06	20/06	04/07
AUCKLAND	SLD	11/05	27/05	12/06	24/06	08/07
YOKOHAMA	SLD	26/05	08/06	27/06	09/07	23/07
KOBE	11/05	28/06	10/06	29/06	11/07	25/07
SHANGHAI	16/05	04/06	16/06	01/07	16/07	01/08
XINGANG	18/05	05/06	18/06	03/07	18/07	03/08

NOTE: TPK=TAI PING KOU, XSK=XIAO SHI KOU, BHK=BAI HE KOU, ZJK=ZHI JIANG KOU
XFK = XI FENG KOU ZGJK = ZHANG JIA KOU GBK = GU BEI KOU

APPENDIX D

4GL Report Generators Comparisan [11]

Characteristic	dBase III +	RBase V	PC-Focus	Oracle	PC-Ingres	Paradox	XDB
Report media specification	Screen-oriented, menu-driven	Screen-oriented, menu-driven	Program-oriented, text file	Program-oriented, text file	Program-oriented, text file	Screen-oriented, menu-driven	Screen-oriented, menu-driven
Encapsulation	Separate entity, poor integration with host system	Separate entity, poor integration with host system	Separate entity, extensive features for standalone use	Separate entity, good integration with host system	Separate entity, poor integration with host system	Separate entity, poor integration with host system	Seperate entity, poor integration with host system
Data source	Exactly one table, base or joined	Exactly one table, base or joined, limited dynamic access to other tables	Exactly one table, base or joined	Exactly one table, base or joined, good dynamic access to other tables	Exactly one table, base or joined	Exactly one table, base or joined	Exactly one table, base or joined
Ordering of input records	Must be done physically before report is started	Must be done physically before report is started	Logical ordering good flexibility, part of report specification	Logical ordering good flexibility, part of report specification	Logical ordering good flexibility, part of report specification	Logical ordering good flexibility, part of report specification	Logical ordering good flexibility, part of report specification
Source of information for report line	Current record of input stream	Current record of input stream, limited dynamic access to other table records	Current record of input stream	Current record of input stream, good dynamic access to other table records	Current record of input stream	Current record of input stream	Current record of input stream
Output media	Adequate	Adequate	Adequate, can also create temporary database	Adequate	Adequate	Adequate	Adequate
Procedural support	None	None	None	Limited support through call to procedural macros	None	None	None
Numeric data editing for display	Limited	Limited	Fair	Fair	Good	Fair	Fair
User-defined report groups with headers and footers	Two levels	10 levels	High, adequate for all practical use	High, adequate for all practical use	High, adequate for all practical use	High, adequate for all practical use	High, adequate for all practical use
Report column definition and placement	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate	Adequate
Arithmetic expression as a column	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Conditional expression in a column	Not supported	Not supported	Supported	Supported	Supported	Not supported	Not supported
Aggregate functions on a column	SUM	SUM	COUNT AVG SUM	COUNT AVG SUM MAX MIN STDV	COUNT AVG SUM MAX MIN	COUNT AVG SUM MAX MIN	COUNT AVG SUM MAX MIN
Conditional aggregation of a column values	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
Arithmetic expression in a column involving aggregation	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
Embedding data values in a header or a footer	Strings and aggregates	Strings and aggregates	Strings and limited aggregate values	Strings, aggregates and limited arithmetic expressions	Strings and aggregates and limited arithmetic expressions	Strings and aggregates	Strings and aggregates
Ease of use	Fair	Fair	Fair	Difficult	Fair	Good	Fair
Ease of learning	Fair	Fair	Fair	Difficult	Fair	Good	Fair

APPENDIX E

Oracle Hardware Requirements [6]

The minimum hardware requirements to run Oracle are:

* A central processing unit equivalent to that of an IBM PC/AT or Oracle-certified compatible (that is, an Intel 80286 or 80386 CPU). The system in this project was done on a 80386 machine, with speed of 25 MHz.

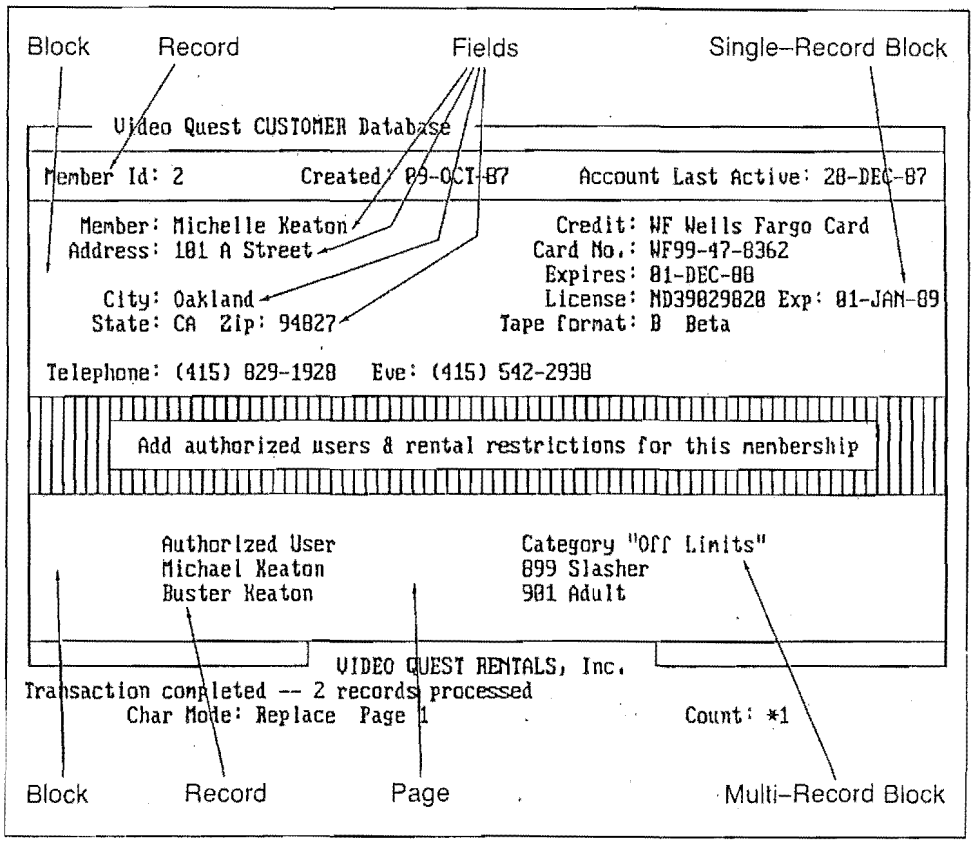
* 640K of random access memory and 896K of extended memory.

* A hard disk with 8.5 megabytes of available storage for real mode and 1.5 megabytes for protected mode.

* A monochrome or colour monitor.

APPENDIX F

SQL*Forms Terminology [15]



Page	The part of the form currently displayed on a full screen at one time. A form can span many pages.
Block	The section of a form that contains a group of related fields. The fields hold data and text that normally reference a single table.
Base Table	The table on which a block is based.
Record	Data from one row of a database table.
Field	A designated area on the screen that can display a value. The value normally corresponds to a value from a column in the database table.
Single-record Block	A block that can display only one record.
Multi-record Block	A block that can display more than one record.

APPENDIX G

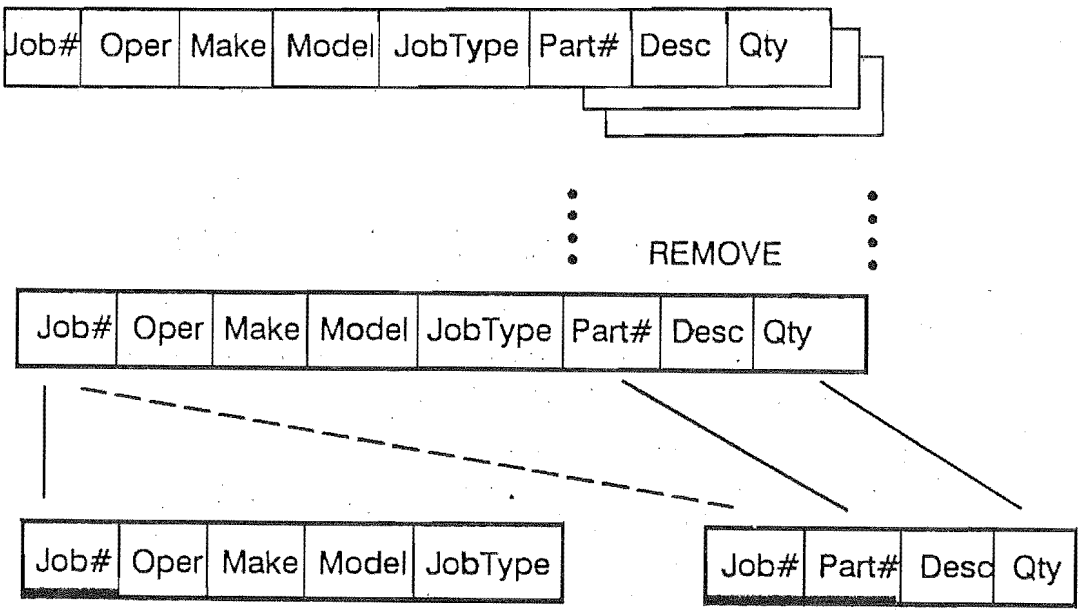
First, Second and Third Normal Form [9]

1st Normal Form

Refers to a collection of data organised into records which contain no repeating groups.

In general, a non-flat file is normalised by converting it to into two or more flat files. The key field of the un-normalised record is added into the key of the repeating group.

So the first step in normalisation is to remove any repeating groups.

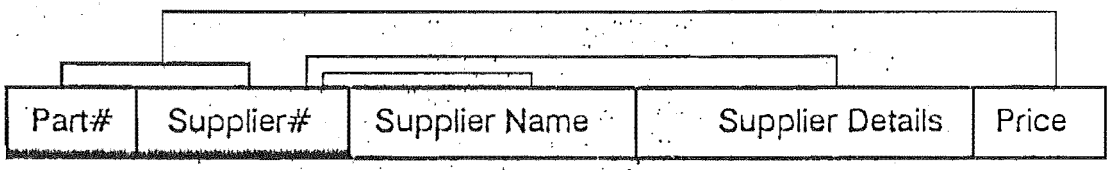


2nd Normal Form

Each attribute of a record is functionally dependent on the whole key of that record.

Where the key of a record is concatenated, that is consists of more than one data field, the record may not be in second normal form. Where a record has no repeating groups and the key consists of single field the record is always in second normal form.

For example, suppose that the garage can purchase a standard type of spark plugs from several suppliers. They will probably be available at different prices. Recording the part number of the plugs will not uniquely identify the source, recording the supplier alone will not identify the part either, since a supplier will probably supply more than one part. In order to identify a plug fully, so that correct price can be found, both the part number and the supplier must be used as a composite key. A typical data structure is shown below



The supplier name and supplier details are dependent only on the supplier part of the key and are not affected by the part number. This is a case of a partial dependency.

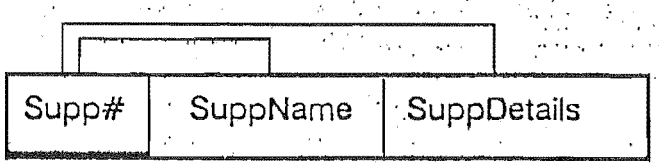
Problems with a non-2NF record

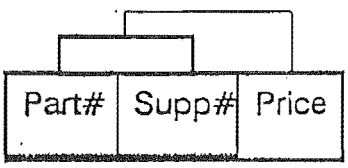
a) Cannot enter details of a supplier until he has supplied something. No part - No key.

b) If the garage runs out of spark plugs from a particular supplier and deletes the product record then all the supplier details could be deleted if that is the only record for that supplier.

c) Updating the supplier details means searching the entire file to find all records which have that supplier in the key.

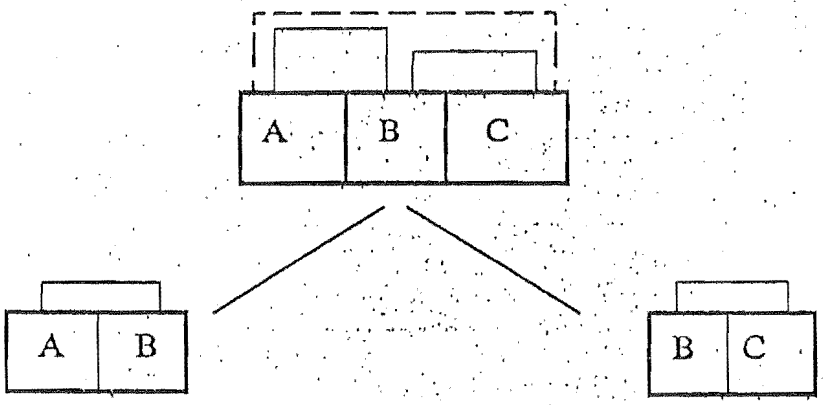
The solution is to split the structure into two separate files .



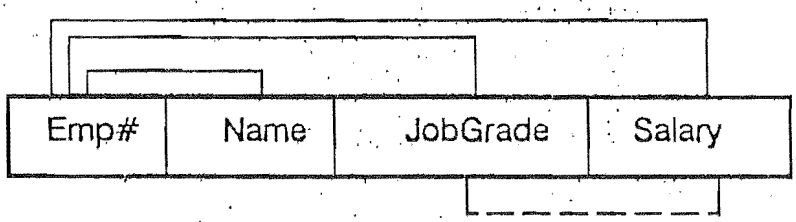


3rd Normal Form

A record may hold a data item which is not itself a key but which functionally determines some other data item in the record. This type of relationship is known as a transitive dependency.



Suppose the garage paid workers according to some national pay agreement based on particular grades of job skills. The record could appear as



Problems with non-3NF records

a) The rate of pay for a grade cannot be recorded unless an employee has that grade, or a dummy employee record is created.

b) If the pay rate changes for a grade every occurence of that grade has to be found and updated.

c) If all employees on a particular grade leave, then all trace of that grade's rate of pay is lost.

These transitive dependencies are removed in the third normal form.

APPENDIX H

Tables in the Database

table	column	attribute	key
			primary (p) foreign (f)
purchase	purno	purchase form number	p
	matno	material number	p,f
	qty	purchased quantity	
	received	received quantity	
pur_sup	purno	purchase form number	p,f
	supno	supplier number	p,f
	datesent	date purchase form is sent	
	daterec	date goods are received	
capacity	process	factory processes	p
	name	process name	
	capa	process maximum capacity	
	rates	process rates per hour	
	req	order's required capacity	
	week1	process available capacity	
	week2	process available capacity	
	week3	process available capacity	
	week4	process available capacity	
	calc1	calculation column	
	calc2	calculation column	
supplier	supno	supplier number	p
	name	supplier's name	
	address	supplier's address	
mrp	matno	material number	p,f
	describe	inventory status	p
	week0	week0 inventory data	
	week1	week1 inventory data	
	week2	week2 inventory data	
	week3	week3 inventory data	
	week4	week4 inventory data	
	calc1	calculation column	
	calc2	calculation column	
performance	dates	date when indicies are recorded	p
	pack	packaging performance index	
	moulding	moulding performance index	
	wastage	material wastage	
	machine1	machine # 1 utilization	
	machine2	machine # 2 utilization	

part	partno	part number	p
	name	part's name	
	cycle	moulding cycle time	
	machno	machine number	f
	mean	mean stock level	
	max	maximum stock level	
	min	minimum stock level	
	dev	deviation from mean level	
	dev0	mean deviation of previous week	
	actual	quantity made	
	calc	calculation column	
material	matno	material number	p
	name	material's name	
	sup	supplier number	f
	cost	material cost	
	lead	lead time of material	
	lot	minimum lot size	
	mean	mean stock level	
	max	maximum stock level	
	min	minimum stock level	
	calc	calculation column	
orders	orderno	order number	p
	customer	customer's name	
	reqdate	delivery date	
	commit	committed order	
	l_o	local or overseas	
	pactime	total packaging time	
	pstart	packaging start date	
	pfinish	packaging finish date	
	packed	ratio of packed products	
	out	delivered order	
container	type	container type	p
	vol	container's volume	
bom	invno	inventory number (parent)	p,f
	have	inventory number (child)	p,f
	qty	quantity of child	
	calc	calculation column	
week	week1	date of Monday of week1	p
	week2	date of Monday of week2	
	week3	date of Monday of week3	
	week4	date of Monday of week4	
product	prodno	product number	p
	vol	product's volume	
	wgt	product's weight	
	calc	calculation column	

dummy1	x	dummy column (for character)	
	y	dummy column (for date)	
	z	dummy column (for number)	
	w	dummy column (for number)	
mps	unitno	unit number	p
	name	unit's name	
	pactime	unit's estimated packaging time	
	overhead	overhead factory cost per unit	
	actual_o	packed unit quantity (overseas)	
	actual_l	packed unit quantity (local)	
	op1	week1 overseas planned quantity	
	oc1	week1 overseas committed quantity	
	lp1	week1 local planned quantity	
	lc1	week1 local committed quantity	
	op2	week2 overseas planned quantity	
	oc2	week2 overseas committed quantity	
	lp2	week2 local planned quantity	
	lc2	week2 local committed quantity	
	op3	week3 overseas planned quantity	
	oc3	week3 overseas committed quantity	
	lp3	week3 local planned quantity	
	lc3	week3 local committed quantity	
	op4	week4 overseas planned quantity	
	oc4	week4 overseas committed quantity	
	lp4	week4 local planned quantity	
	lc4	week4 local committed quantity	
	calc1	calculation column	
	calc2	calculation column	
	op_1	temporary data storage	
	oc_1	temporary data storage	
	lp_1	temporary data storage	
	lc_1	temporary data storage	
	op_2	temporary data storage	
	oc_2	temporary data storage	
	lp_2	temporary data storage	
	lc_2	temporary data storage	

APPENDIX I

SQL*Menu Documentation

Appendix A

Summary Menu Application Information

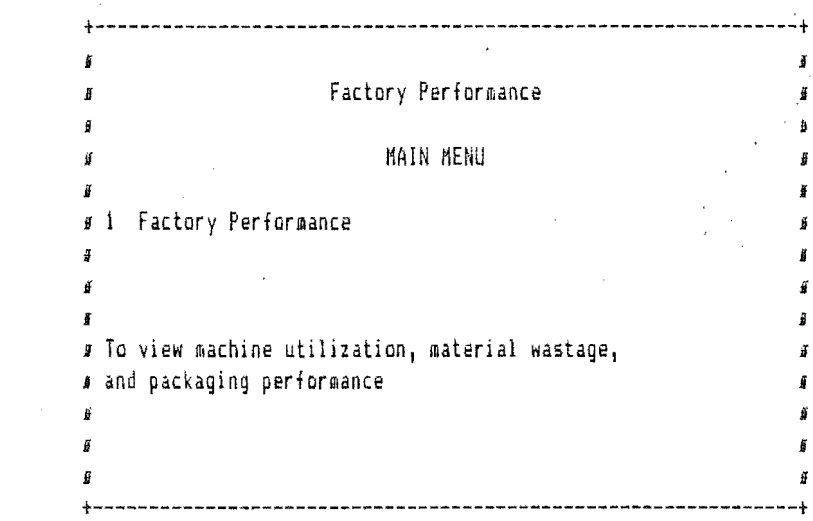
A.1 Application Information

Application	PERFOM
Creation date	06-APR-90
Creator	SYSTEM
Last release date	
Menu directory	
Version	1.1

A.2 Menu overview.

PERFOM

A.3 PERFOM



A.3.1 Option number 1

Work class range 20 - 20
Command type Runforms (IAP)

Command line
============
runform -m performance &un/&pw

A.4 Work Class 20

manager

Username	BG Menu	OS Command	Debug
SYSTEM	Y	Y	Y

Appendix A

Summary Menu Application Information

A.1 Application Information

Application	FACTRE
Creation date	04-APR-90
Creator	SYSTEM
Last release date	
Menu directory	
Version	1.1

A.2 Menu overview.

FACTRE

A.3 FACTRE

```
+------------------------------------------------------------+
#                                                            #
#                        FACTORY DATA                        #
#                                                            #
#                         MAIN MENU                          #
#                                                            #
# 1  Input of Production Data                                #
# 2  Input of Inward/Outward Goods Data                      #
# 3  Material Purchase Proposals                             #
#                                                            #
#                                                            #
#                                                            #
# To input/update production and material data               #
#                                                            #
#                                                            #
#                                                            #
#                                                            #
+------------------------------------------------------------+
```

A.3.1 Optio

Work class range 10 - 20
Command type Runforms (IAP)

Command line
============
runform -m factory &un/&pw

A.3.2 Option number 2

Work class range 10 - 20
Command type Runforms (IAP)

Command line
============
runform -m control &un/&pw

A.3.3 Option number 3

Work class range 10 - 20
Command type Runforms (IAP)

Command line
============
runform -m reorder &un/&pw

Summary Menu Application Information 07-JUN-90 Page A-3

A.4 Work Class 10

worker

Username	BG Menu	OS Command	Debug
CLERK	N	N	N

Summary Menu Application Information 07-JUN-90 Page A-4

A.5 Work Class 20

manager

Username	BG Menu	OS Command	Debug
SYSTEM	Y	Y	Y

Appendix A

Summary Menu Application Information

A.1 Application Information

Application	NEWORD
Creation date	04-APR-90
Creator	SYSTEM
Last release date	
Menu directory	
Version	1.1

A.2 Menu overview.

NEWORD

A.3 NEWORD

```
+-------------------------------------------------------------+
#                                                             #
#                  Create  New  Orders                        #
#                                                             #
#                       MAIN MENU                             #
#                                                             #
# 1  Create New Orders                                        #
# 2  Load New Orders                                          #
#                                                             #
# COMPANY CONFIDENTIAL                                        #
#                                                             #
# To create new orders and check for vol.,wgt.,and material   #
# cost.                                                       #
#                                                             #
#                                                             #
#                                                             #
+-------------------------------------------------------------+
```

A.3.1 Option number 1

Work class range 10 - 20
Command type Runforms (IAP)

Command line
============
runform -m creorder &un/&pw

A.3.2 Option number 2

Work class range 10 - 20
Command type Runforms (IAP)

Command line
============
runform -m capacity &un/&pw

Summary Menu Application Information 07-JUN-90 Page A-3

A.4 Work Class 10

non-management employee

Username	BG Menu	OS Command	Debug
CLERK	N	N	N

Summary Menu Application Information 07-JUN-90 Page A-4

A.5 Work Class 20

manager

Username	BG Menu	OS Command	Debug
SYSTEM	Y	Y	Y

Appendix A

Summary Menu Application Information

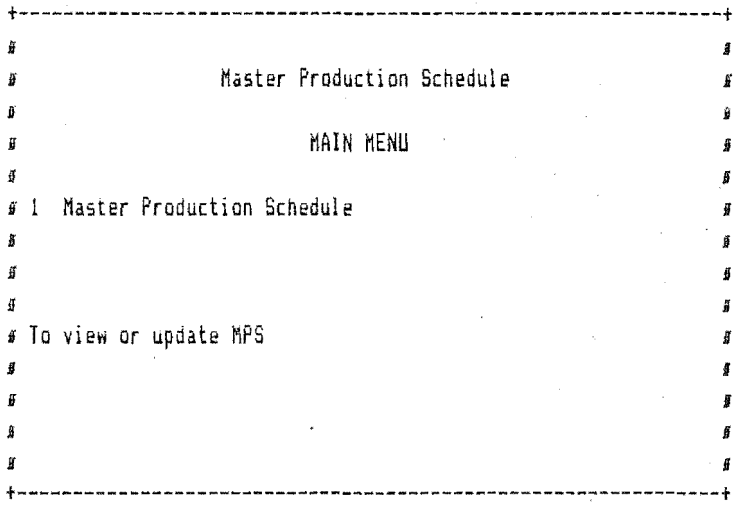
A.1 Application Information

Application	MASTER
Creation date	04-APR-90
Creator	SYSTEM
Last release date	
Menu directory	
Version	1.1

A.2 Menu overview.

MASTER

A.3 MASTER



```
A.3.1 Option number    1

  Work class range  20  -  20
  Command type      Runforms (IAP)

  Command line
  ============
  runform -m master &un/&pw
```

```
Summary Menu Application Information          07-JUN-90   Page A-3

A.4 Work Class   20

  manager

  Username                BG Menu       OS Command     Debug

  SYSTEM                  Y             Y              Y
```

1. DEBUG MODE (DBM). This privilege determines whether or not the user can run SQL*Menu in DEBUG MODE, a utility that isolates menu application problems by displaying the command line behind a menu option when it is chosen.
2. OPERATING SYSTEM COMMANDS (OSC). This privilege determines whether or not the user can run operating system commands within a SQL*Menu application.
3. BACKGROUND MENU (BGM). This determines whether or not the user can access the BACKGROUND Menu, a hidden menu that allows privileged users to run various ORACLE tools and reports, frequently used commands, and even non-ORACLE applications or programs from within the SQL*Menu application. The ability to run SQL*Forms (Design) or SQL*Plus within the SQL*Menu environment, for example, can save a designer valuable time while debugging forms and reports [7]

APPENDIX J

Detail Description of the Forms

This appendix describes the major functions and computations that are involved in the forms. The descriptions are accompanied by the appropriate database tables and include references to the triggers that are responsible for the computations. The references (K1, K2, etc.) refer to the specific trigger statements in appendix K.

The 'explosion' process is illustrated by these symbols:

This illustrates the 'explosion' of items that are higher in the product structure into lower level items.

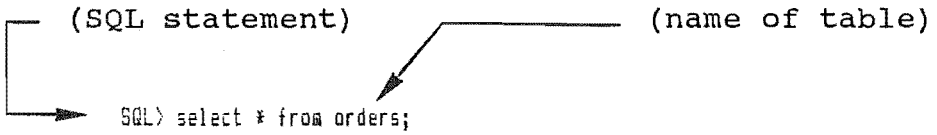
This illustrates the summing up of the exploded items' quantities according to their inventory numbers.

MASTER form

Checking the validity of the schedule

The major function performed in the MASTER form is the checking of the production schedule to determine its validity. This include checking the availability of factory capacity and materials to support the schedule.

Test data of committed orders is inserted into the orders and ord_prod tables, as shown below.




```
SQL> select * from orders;

ORDERNO CUSTOMER   REQDATE   CO L PACTIME PSTART    PFINISH    PACKED O
------- ---------- --------- -- - ------- --------- --------- ------- -
    107 neworld    29-MAY-90 y  1    .215 28-MAY-90 28-MAY-90
   2002 france     29-MAY-90 y  o      .2 28-MAY-90 28-MAY-90
    108 w-mart     06-JUN-90 y  1    .044 04-JUN-90 04-JUN-90
   2003 england    06-JUN-90 y  o    .083 04-JUN-90 04-JUN-90

SQL> select * from ord_prod;

ORDERNO  PRODNO    PQTY    PACK    CALC
------- ------- ------- ------- -------
    107   19533       5
   2002   19525       8
    108   19500       4
   2003   19500       3
   2003   19525       2
```

(Some column names in the orders table have been truncated, refer to appendix H)

The units from these orders show up in the committed or 'C' columns of the MPS (figure 16). With demand predictions on hand, the production is scheduled. Assume

that the schedule, as shown in figure 16, is the finalised one.

Checking the factory capacities

To check the validity of the schedule, the user enters the 'y' character in the field at the top of page 2 (figure 17). This trips a trigger ^{K1} which does a series of computations. Samples of the calculations that are involved are shown below.

The values in the AVAIL fields of page 2 (figure 17) are the results of the calculation:

maximum allowable capacity - planned capacity.

Checking the packaging capacity

Calculation of the available capacity for the packaging process:

= maximum capacity - total planned packaging time

= maximum capacity - sum (planned unit quantity * packaging time)

available capacity (for week 28-MAY-90)

= 5 - (90*20 + 90*25 +90*30)/3600

= 3.125 hr

For this test run, the maximum allowable capacity for all the production processes is fixed at 5 hours. The planned quantity of 90 is the total quantity, i.e.

overseas quantity + local quantity (see figure 16). The packaging time for the units is recorded in the pactime column, in the mps table:

```
SQL> select unitno,pactime
  2  from mps;

 UNITNO PACTIME
------- -------
   9500      20
   9505      25
   9520      30
```

Checking the moulding processes

Explosion of the planned units (for week 28-MAY-90):

unitno	qty
9500	90
9505	90
9520	90

partno	qty
500	90
501	90
500	90
511	90
520	90

partno	qty
500	180
501	90
511	90
520	90

The relations between the units and the parts are given in the bom table:

INVNO	HAVE	QTY
9500	500	1
9500	501	1
9505	500	1
9505	511	1
9520	520	1

The MPS specifies the quantities of units that are needed in the packaging process. This in turn determines the output of the moulding process. The calculation of

the moulding output also has to take into account the stock levels of the parts.

For example, 180 units of part 500 are needed in the packaging process for the present week (28-MAY-90). The stock level for part 500, at the end of previous week:

= mean level + deviation from mean,
at the end of previous week

= mean + dev0 (columns of the part table)

= 100 + 5 units (boxed figures below)

```
SQL> select partno,name,cycle,
  2  mean,max,min,dev0
  3  from part;
```

PARTNO	NAME	CYCLE	MEAN	MAX	MIN	DEV0
500	COVER-00	40	[100]	125	75	[5]
501	BASE-00	20	100	125	75	5
511	BASE-05	25	100	125	75	5
520	COL-B	50	100	125	75	5

The planned output for part 500 for the present week:

= 180 - dev0 value

= 180 - 5 units.

The adjustment of the output, by the dev0 value, is necessary for maintaining the stock as close to the mean level as possible.

The adjustment is reflected in the calculation of the available capacity for the moulding processes:

available capacity for machine # 1
(moulds parts 500 and 501)

= maximum capacity - total planned moulding time

= maximum capacity - sum (planned output of part * moulding cycle time)

= 5 - ((180-5)*40 + (90-5)*20))/3600

= 2.583 hr

available capacity for machine # 2
(moulds parts 511 and 520)

= 5 - ((90-5)*25 + (90-5)*50)/3600

= 3.229 hr

The moulding cycle time is recorded in the cycle column, in the part table:

```
SQL> select partno,machno,cycle,calc
  2  from part;

 PARTNO  MACHNO   CYCLE
------- ------- -------
    500       1      40
    501       1      20
    511       2      25
    520       2      50
```

Checking the ordering of materials

The purchased materials can be grouped into three categories based on how their requirements are derived:

i) Plastic raw materials (matno 1070 - 1090)

Their requirements are derived from the planned output of parts.

ii) Packaging materials (matno 1040 - 1060)

Some examples of these materials are plastic bags, labels, gift boxes and bar code stickers. Their requirements are derived from the number of units that will be packaged. The packaging of each unit may include one or more of these materials.

iii) Outer carton boxes (matno 1010 - 1030)

They are also packaging materials but their requirements are associated with the quantities of products that will be packaged.

The computation of the requirements for matno 1070 - 1090 are described below.

Explosion of the planned output of parts (for week 28-MAY-90):

partno	qty	matno	qty (kg)	matno	qty (kg)
500	175	1070	175*0.4 *1.15	1070	80.5
520	85	1080	85 *0.3 *1.15	1080	29.325
501	85	1090	85 *0.15*1.15	1090	39.1
511	85	1090	85 *0.25*1.15		

The first multiplication factor, in the calculations above, is the quantity of the material used in moulding each part (see bom table below). The second multiplication factor of 1.15 is for material wastage.

INVNO	HAVE	QTY
500	1070	.4
520	1080	.3
501	1090	.15
511	1090	.25

The material requirements are inserted into the 'req' rows in the MRP table. The 'dev' rows record the deviation from stock mean levels. Expected deliveries are recorded in the 'del' rows.

matno	describe	week0	week1	week2	week3	week4
1070	req		80.5	87.4	59.8	64.4
1070	dev		-5	0	0	0
1070	del		70	0	0	0
1070	or		15.5	87.4	59.8	64.4

The values in the order release ('or') row are computed as follows:

values in the 'req' row
minus values in the 'dev' row
minus values in the 'del' row
gives values in the 'or' row

Orders with quantities that are less than the minimum lot size are considered not economical to order. These order quantities are added to the orders in the following week. The release of the orders is arrived at

by offsetting for lead time:

matno	describe	week0	week1	week2	week3	week4
1070	or	15.5	87.4	59.8	64.4	0

Week1 is the present week. An order release in the week0 field indicates that the order should have been released earlier than the present week in order to support the current schedule (e.g. material 1070 above).

The MPS is rescheduled by reducing the planned overseas and local units in week1 (28-MAY-90) (figure J1). The resulting MRP shows a zero value in the week0 field for material 1070 and all the other materials (figure J2). The new schedule is now a valid one.

MASTER PRODUCTION SCHEDULE

		28-MAY-90				04-JUN-90				11-JUN-90				18-JUN-90			
		OVERSEA		LOCAL		OVERSEA		LOCAL		OVERSEA		LOCAL		OVERSEA		LOCAL	
UNITNO	ACT	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C
9500__	-80	40_	0__	40_	20_	55_	6__	40_	8__	50_	0__	15_	0__	55_	0__	15_	0__
9505__	-80	40_	0__	40_	15_	55_	0__	40_	0__	50_	0__	15_	0__	55_	0__	15_	0__
9520__	-80	40_	24_	40_	0__	55_	6__	40_	0__	50_	0__	15_	0__	55_	0__	15_	0__
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SCHEDULE NEW WEEK ___

Figure J1 Page 1 of MASTER form

```
+-----------------------MATERIAL-REQUIREMENT-----------------------+
#                                                                  #
#  MATNO   DESCRIBE  WEEK0   WEEK1   WEEK2   WEEK3   WEEK4        #
#  1070__  req_____  ______  71.3__  87.4__  59.8__  64.4__       #
#  1070__  dev_____  ______  -5____  0_____  0_____  0_____       #
#  1070__  del_____  ______  70____  0_____  0_____  0_____       #
#  1070__  or______  0_____  93.7__  59.8__  64.4__  0_____       #
#  ______  _______   ______  ______  ______  ______  ______       #
#  1080__  req_____  ______  25.875  32.775  22.425  24.15_       #
#  1080__  dev_____  ______  .1____  0_____  0_____  0_____       #
#  1080__  del_____  ______  35____  35____  0_____  0_____       #
#  1080__  or______  0_____  10.975  24.15_  0_____  0_____       #
#  ______  _______   ______  ______  ______  ______  ______       #
#  1090__  req_____  ______  34.5__  43.7__  29.9__  32.2__       #
#  1090__  dev_____  ______  5_____  0_____  0_____  0_____       #
#  1090__  del_____  ______  45____  0_____  0_____  0_____       #
#  1090__  or______  0_____  28.2__  29.9__  32.2__  0_____       #
+------------------------------------------------------------------+
#  EXIT:                                                           #
#                                                                  #
#  RESTORE COMMITED CAPACITY DATA  ___                             #
+------------------------------------------------------------------+
```

Figure J2 Page 3 of MASTER form

```
+--------------------ORDER'S-PARTICULARS---------------------------+
#                                                                  #
#          L/O   I____     REQDATE   07-JUN-90                     #
#                                                                  #
#         ORDERNO 109__      CUSTOMER farmer's__                   #
#                                                                  #
+------------------------------------------------------------------+

                INPUT PRODUCT AND QUANTITY
+------------------------------------------------------------------+
#    ORDERNO  PRODNO   NAME                          PQTY          #
#    109____  19533__  LUNCH BOX S10*4,S11*3_____    10_______     #
#    109____  19525__  COLANDER-BIG * 3__________    15_______     #
#    109____  19500__  LUNCH BOX S10 * 2_________    10_______     #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
#    _______  _______  ----------------------------  ----------    #
+------------------------------------------------------------------+
```

Figure J3 Page 1 of CREORDER form

CREORDER form

When orders are received from the customers, the processing starts with the CREORDER form. Particulars of the orders are entered in page 1 (figure J3). The computation ^{K2} to process the data is given below.

Calculating order's volume and weight

The volume and weight data is kept in the product table, as shown below.

```
SQL> select * from product;

 PRODNO NAME                          VOL     WGT    CALC
------- ------------------------ ------- ------- -------
  19500 LUNCH BOX S10 * 2              3       2      10
  19525 COLANDER-BIG * 3               4       6      15
  19533 LUNCH BOX S10*4,S11*3          5       7      10
```

The order's volume

= sum (product quantity * product volume)

= 10*3 + 15*4 + 10*5

= 140 m^3

Next, the number of shipping containers needed for this volume is calculated.

The order's weight

= sum (product quantity * product weight)

= 10*2 + 15*6 + 10*7

= 180 kg

Cost calculation

Explosion of the order's products:

prodno	qty
19533	10
19525	15
19500	10

unitno	qty
9500	10*4
9505	10*3
9520	15*3
9500	10*2

unitno	qty
9500	60
9505	30
9520	45

invno	qty
19533	10
19525	15
19500	10

matno	qty
1030	10*1
1020	10*1
1010	15*1
1010	10*1

matno	qty
1030	10
1020	10
1010	25

Explosion of the order's units:

unitno	qty
9500	60
9505	30
9520	45

partno	qty
500	60*1
501	60*1
500	30*1
511	30*1
520	45*1

partno	qty
500	90
501	60
511	30
520	45

unitno	qty
9500	60
9505	30
9520	45

matno	qty
1040	60*1
1050	60*1
1040	30*1
1060	30*1
1040	45*1

matno	qty
1040	135
1050	60
1060	30

Explosion of the order's parts:

matno	qty
500	90
501	60
511	30
520	45

matno	qty
1070	90*.4*1.15
1090	60*.15*1.15
1090	30*.25*1.15
1080	45*.3*1.15

matno	qty
1070	41.4
1090	18.975
1080	15.525

The results of the explosions are shown in the calc column of the bom table:

INVNO	HAVE	QTY	CALC
9500	1050	1	60
9505	1040	1	30
9505	1060	1	30
9520	1040	1	45

INVNO	HAVE	QTY	CALC
19500	9500	2	20
19533	9500	4	40
19533	9505	3	30
19525	9520	3	45
9500	500	1	60
9500	501	1	60
9505	500	1	30
9505	511	1	30
9520	520	1	45
19500	1010	1	10
19533	1030	1	10
19533	1020	1	10
19525	1010	1	15
500	1070	.4	41.4
520	1080	.3	15.525
501	1090	.15	10.35
511	1090	.25	8.625
9500	1010	.04	0
9500	1030	.05	0
9505	1020	.02	0
9520	1010	.01	0
9500	1040	1	60

Cost calculations:

material cost
= sum (material quantity * material cost)
= (matno 1010 - 1030) 25*.15 + 10*.15 + 10*.15 +
(matno 1040 - 1060) 135*.07 + 60*.01 + 30*.02 +
(matno 1070 - 1090) 41.4*2 + 18.98*2 + 15.53*2
= $169.22

packaging cost
= order's packaging time * rates
= sum (unit quantity * packaging time) * rates
= (60*20 + 30*25 + 45*30)/3600 * 10
= $9.17

moulding cost
= order's moulding time * rates
= sum (part quantity * moulding cycle time * rates)
= (90*40 + 60*20)/3600 * 10 +
(30*25 + 45*50)/3600 * 10
= $21.66

```
overhead cost
=     sum (unit quantity * overhead)
=     60*.2 + 30*.2 + 45*.2
=     $27.00
```

The material, unit and part quantities in the calculations come from the explosion of order 109, as shown above. The rest of the data used in the calculations is kept in these columns:

```
SQL> select matno,cost from material;

  MATNO    COST
------- -------
   1010     .15
   1020     .15
   1030     .15
   1070       2
   1080       2
   1090       2
   1040     .07
   1050     .01
   1060     .02
```

```
SQL> select process,name,rates
  2  from capacity;

PROCESS NAME           RATES
------- ------------ -------
     90 pack_lo
     91 pack_os           10
      1 machine1          10
      2 machine2          10
```

```
SQL> select unitno,pactime,overhead,calc1
  2  from mps;

 UNITNO PACTIME OVERHEAD   CALC1
------- ------- -------- -------
   9500      20       .2      60
   9505      25       .2      30
   9520      30       .2      45
```

```
SQL> select partno,machno,cycle,calc
  2  from part;

 PARTNO  MACHNO   CYCLE    CALC
------- ------- ------- -------
    500       1      40      90
    501       1      20      60
    511       2      25      30
    520       2      50      45
```

CAPACITY form

The order processing continues in the CAPACITY form.

Calculating the order's required capacity

The required capacity of order 109 is calculated by inserting 'y' into the CALC field in page 1 (figure 23):

required packaging capacity
= sum (required units * packaging time)
= (60*20 + 30*25 +45*30)/3600
= 0.917 hr

required machine # 1 capacity
= sum (required parts * moulding cycle time)
= (90*40 + 60*20)/3600 [parts 500 and 501]
= 1.333 hr

required machine # 2 capacity
= (30*25 + 45*50)/3600 [parts 511 and 520]
= 0.833 hr

The required units and parts in the calculations come from the explosion of order 109, as shown earlier.

Loading order into the schedule [K3]

The top of page 1 (figure 23) displays the available capacity in the factory, which is the uncommitted capacity (i.e. maximum allowable capacity - committed capacity). The user can choose and load the order into the weekly period of the schedule which

1) has sufficient capacity to support the making of the order, and

2) will allow the packaging of the order to be completed before the expected delivery date.

Order 109 is loaded into the week1 (28-MAY-90) and week2 (4-JUN-90) schedules (bottom of figure 23). This means that the production of order 109 may start in week1 and finish in week2. The units to be packaged in each week is automatically calculated and displayed in page 5 (boxed figures in figure J4). Page 5 displays also the weekly periods of the MPS, which the order's units have been loaded into (order 109 is loaded into the local section of the MPS).

```
+----------------------------------PRODUCTION-------------------------------+
#                START              FINISH           DELIVERY               #
#                28-MAY-90          04-JUN-90        07-JUN-90              #
+-------------------------+-------------------------------+-----------+
                          #     UPDATE MPS y___           #
                          #                               #
                          #                               #
+--------+----------------+---------------MASTER-SCHEDULE-+-------------+
#        #                #                               #             #
#        # ORDER'S QTY    #      28-MAY-1990              #  04-JUN-1990  #
#        +--------+-------+                               #             #
#        # FIRST  # SECOND #  OVERSEAS    LOCAL           # OVERSEAS  LOCAL #
# UNITNO # WEEK   # WEEK   #  PLN  COMM PLN  COMM         # PLN  COMM PLN COMM #
# 9500__ # 40__   # 20__   #  40__ 0___ 60__ 60__         # 55__ 6___ 40__ 28__ #
# 9505__ # 20__   # 10__   #  40__ 0___ 35__ 35__         # 55__ 0___ 40__ 10__ #
# 9520__ # 30__   # 15__   #  40__ 24__ 30__ 30__         # 55__ 6___ 40__ 15__ #
# ______ # ____   # ____   #  ____ ____ ____ ____         # ____ ____ ____ ____ #
# ______ # ____   # ____   #  ____ ____ ____ ____         # ____ ____ ____ ____ #
#        #        #        #                             #                   #
+--------+--------+--------+-----------------------------+-------------------+
#       RECALCULATE CAPACITY AND MATERIAL REQ. ___                            #
+-----------------------------------------------------------------------------+
```

Figure J4 Page 5 of CAPACITY form

Checking material availability and factory capacity

The loading of an order into the schedule may lead to changes to the planned material requirements. In the case of order 107, it leads to an order release in the week0 period (figure J5). This is reported on page 4 of the form (figure J6). This means that material 1070 may not be ordered in time to support the production of order 107.

Displayed at the bottom of page 4 are the uncommitted capacities after the loading. The user checks that there are no negative values in them.

If the results, as presented in page 4, is not satisfactory the user may

1) reschedule the MPS, or
2) load order into the other periods of the schedule.

The user checks also that the estimated completion date in producing the order is before the delivery date (top of page 5).

```
+-----------------------MATERIAL-REQUIREMENT-----------------------+
#                                                                  #
#  MATNO    DESCRIBE   WEEK0    WEEK1    WEEK2    WEEK3    WEEK4  #
#  1070__   req_____   ______   78.2__   87.4__   59.8__   64.4__ #
#  1070__   dev_____   ______   -5____   0_____   0_____   0_____ #
#  1070__   del_____   ______   70____   0_____   0_____   0_____ #
#  1070__   or______   13.2__   87.4__   59.8__   64.4__   0_____ #
#  ______   ________   ______   ______   ______   ______   ______ #
#  1080__   req_____   ______   22.425   32.775   22.425   24.15_ #
#  1080__   dev_____   ______   .1____   0_____   0_____   0_____ #
#  1080__   del_____   ______   35____   35____   0_____   0_____ #
#  1080__   or______   0_____   0_____   31.675   0_____   0_____ #
#  ______   ________   ______   ______   ______   ______   ______ #
#  1090__   req_____   ______   36.512   43.7__   29.9__   32.2__ #
#  1090__   dev_____   ______   5_____   0_____   0_____   0_____ #
#  1090__   del_____   ______   45____   0_____   0_____   0_____ #
#  1090__   or______   0_____   30.212   29.9__   32.2__   0_____ #
+------------------------------------------------------------------+
#  EXIT:                                                           #
#                                                                  #
#  RESTORE COMMITED CAPACITY DATA  ___                             #
+------------------------------------------------------------------+
```

Figure J5 Page 3 of MASTER form

```
   CHECK MATERIAL  y__                ___  (press enter to
                                           go to master form)
+-------------MATERIAL-SHORTAGE----------------------+
#                                                    #
#              MATERIAL   QUANTITY                   #
#              1070____   13.2____                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
#              ________   ________                   #
+----------------------------------------------------+
+----------AVAILABLE-CAPACITY-AFTER-LOADING----------+
#                                                    #
#          PROCESS        1ST WK   2ND WK           #
#          PACK (LOCAL)   0_____   .483__           #
#          PACK (O.S. )   3.974_   4.083_           #
#          MACHINE #1     3.692_   4.322_           #
#          MACHINE #2     4.11__   4.64__           #
+----------------------------------------------------+
```

Figure J6 Page 4 of CAPACITY form

FACTORY form

Pages 1 and 2 are used to enter the moulding processes and packaging data. The data entry is to be done at the end of every shift.

Moulding processes data entry

The data entered for parts moulded by machines # 1 and # 2 is shown in figures 29 and J7, respectively. On completion of this data entry, the user updates [K4] the stock levels by inserting 'y' in the UPDATE STOCK field (at the bottom of page 1).

```
                MOULDING MACHINES DATA INPUT
+-------------------------------------------------------------+
#                                                             #
#    PROCEED? y__                    MACHINE NO. 2___         #
#                                                             #
#             MACHINE COUNTER                                 #
#    PARTNO   START    FINISH   REJECTS   TOTAL               #
#    511_____ 2050____ 2070____ 3_______ 17_____             #
#    520_____ 3020____ 3045____ 2_______ 23_____             #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#    -------- -------- -------- --------- --------           #
#                                                             #
+-------------------------------------------------------------+
#                                                             #
#              UPDATE STOCK y__                               #
+-------------------------------------------------------------+
```

Figure J7 Page 1 of FACTORY form

The part table, before and after the updating, is shown below,

```
SQL> select partno,dev,actual,calc
  2  from part;
```

PARTNO	DEV	ACTUAL	CALC
500	5	0	0
501	5	0	0
511	5	0	0
520	5	0	0

(before)

PARTNO	DEV	ACTUAL	CALC
500	23	18	18
501	32	27	27
511	22	17	17
520	28	23	23

(after)

The computations involved are:

DEV = DEV + CALC
ACTUAL = ACTUAL + CALC

For example, for partno 500:

DEV = 5 + 18 = 23 units
ACTUAL = 0 + 18 = 18 units

The values in the calc column, after the updating, correspond to those in the TOTAL field in page 1 (i.e. quantities of parts that are moulded in the shift). The actual column records the quantities of parts moulded in a week. At the beginning of each week the actual column is zeroed. Dev column records the deviations from the mean level (i.e. DEV + MEAN gives the actual stock level).

The stock levels of the plastic materials are also updated. The quantities of materials that are used in the shift:

partno	qty	matno	qty (kg)		
500	18	1070	18*.4*1.15	=	8.280
501	27	1090	27*.15*1.15	=	4.658
511	17	1090	17*.25*1.15	=	4.888
520	23	1080	23*.3*1.15	=	7.935

The deviations are then,

matno	dev (kg)		
1070	-5 - 8.280	=	-13.28
1080	0.1 - 7.935	=	-7.835
1090	5 - 4.658 - 4.888	=	-4.546

The initial deviation values, i.e. -5, 0.1 and 5, are from the 'dev' rows of the mrp table (figure J5).

Packaging process data entry

The quantities of packed products, in the shift, are entered in the PACKED field in page 2 (figure 30).

The updating computation ^{K5} due to this data entry is described below.

The products packed are 'exploded' to their units, parts and materials. The end results of this explosion are shown in the calc column in the bom table below.

INVNO	HAVE	QTY	CALC
19500	9500	2	12
19533	9500	4	20
19533	9505	3	15
19525	9520	3	21
9500	500	1	32
9500	501	1	32
9505	500	1	15
9505	511	1	15
9520	520	1	21
19500	1010	1	6
19533	1030	1	5
19533	1020	1	5
19525	1010	1	7
9500	1040	1	32
9500	1050	1	32
9505	1040	1	15
9505	1060	1	15
9520	1040	1	21

The dev column in the part table is updated to reflect the quantities of parts that were taken out of the stock during the packaging process.

i.e. DEV = DEV - CALC (bom table)

For example, for partno 500:

DEV = 23 - [32 + 15 (boxed figures above)]
= -24 units

```
SQL> select partno,dev,max,min,mean
  2  from part;
```

PARTNO	DEV	MAX	MIN	MEAN
500	-24	125	75	100
501	0	125	75	100
511	7	125	75	100
520	7	125	75	100

Similarly, the materials deviation levels are updated to reflect the quantities used in the packaging process. The deviation levels after the updating are shown below.

```
SQL> select matno,week1 from mrp
  2  where describe='dev'
  3  order by matno;

  MATNO   WEEK1
------- -------
   1010      -3
   1020     -10
   1030      -1
   1040     -78
   1050     -22
   1060     -30
   1070  -13.28
   1080  -7.835
   1090  -4.545
```

Other tables are also updated. The pack column in the ord_prod table, which records the quantities of products that have been packed, is updated (boxed figures below).

```
SQL>
SQL> select * from ord_prod;

ORDERNO  PRODNO    PQTY    PACK    CALC
------- ------- ------- ------- -------
    107   19533       5
   2002   19525       8
    108   19500       4
   2003   19500       3
   2003   19525       2
    109   19533      10       5       5
    109   19525      15       7       7
    109   19500      10       6       6
   2004   19500      15
   2004   19533      15
   2004   19525      15
```

The packed column in the orders table, which records the ratio of orders that has been packed, is updated (boxed figure below).

```
SQL> select * from orders;

ORDERNO CUSTOMER   REQDATE   CO L PACTIME PSTART    PFINISH    PACKED O
------- ---------- --------- -- - ------- --------- --------- ------- -
    107 neworld    29-MAY-90 y  l    .215 28-MAY-90 28-MAY-90
   2002 france     29-MAY-90 y  o      .2 28-MAY-90 28-MAY-90
    108 w-mart     06-JUN-90 y  l    .044 04-JUN-90 04-JUN-90
   2003 england    06-JUN-90 y  o    .083 04-JUN-90 04-JUN-90
    109 farmer's   07-JUN-90 y  l    .917 28-MAY-90 04-JUN-90    .498
   2004 u.s.a.     13-JUN-90 y  o   1.188 11-JUN-90 12-JUN-90
```

The actual_l and actual_o columns in the MPS table, which record the quantities of units that have been packed for the local and overseas orders, are updated (boxed figures below).

```
SQL> select unitno,calc1,actual_l,actual_o from mps
  2  where unitno between 9500 and 10000;

 UNITNO   CALC1 ACTUAL_L ACTUAL_O
------- ------- -------- --------
   9500      32       32        0
   9505      15       15        0
   9520      21       21        0
```

Stock levels and WIP reports

Pages 3 and 4 (figure 31 and 32) allow the user to query the system on whether or not any part or material stock levels lie outside their maximum or minimum levels.

The CURRENT fields in the two pages show the current stock levels (i.e. DEV + MEAN). The message at the bottom of page 3, in figure 31, indicates that there is no stock level that is outside its maximum or minimum levels. Page 4, in figure 32, indicates that stock levels of materials 1040 and 1060 have gone below their minimum levels.

Page 5 shows the work-in-process (figure 33). The data, as shown in this page, is from the orders table. The query function, in which selected records are retrieved from the tables and displayed in the pages, is built by using the SPECIFY DEFAULT ORDERING mode in the forms. The query statements used in pages 3 and 5 are shown in figures J8 and J9.

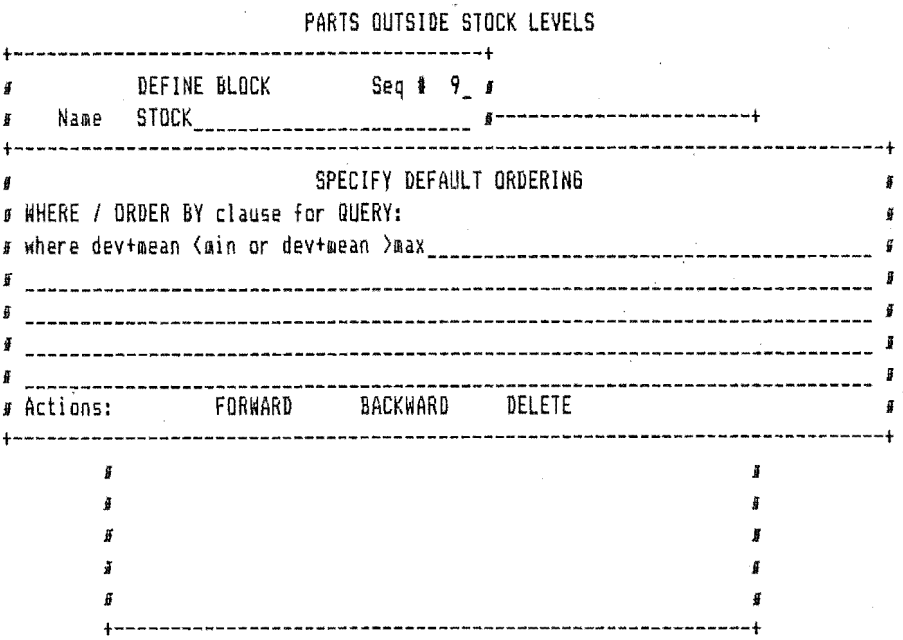


Figure J8 Query statement for stock level

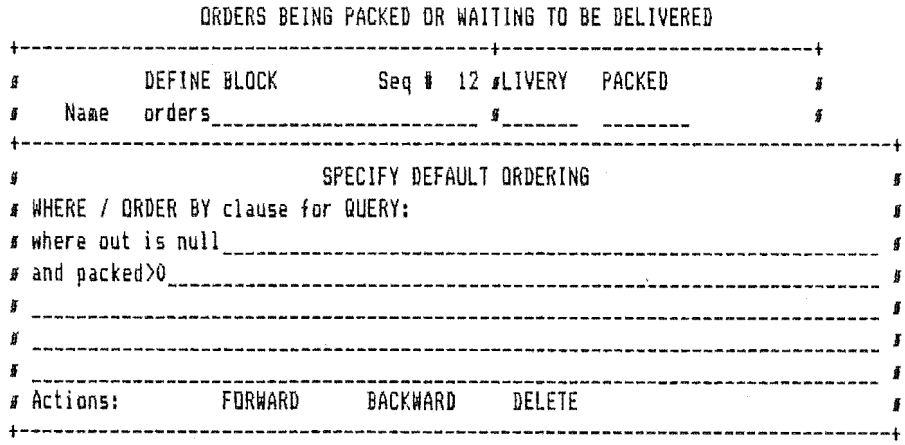


Figure J9 Query statement for WIP

REORDER form

This form is used to order materials when the stock levels are low.

Ordering proposals

Querying at the top half of page 1 (figure J10 (a)) brings up the materials that have order releases in the present week. It proposes to the user the materials and their quantities to order and the suppliers to order from.

Amend proposals

The bottom half of page 1 allows the user to change these proposals. By entering 'y' to the 'PROCEED....' field, a new purchase number and one of the material order proposals, are displayed (figure J10 (a)). (All the order proposals will be displayed, one after the other.) The user has two options with the displayed proposal:

1) accept the proposal as it is, or
2) amend the proposal (i.e. change the order quantity and/or nominate a different supplier) and accept the amended version (figures J10 (a - f)).

The purchase, pur_sup and mrp tables are updated [K6] when the proposals are 'accepted':

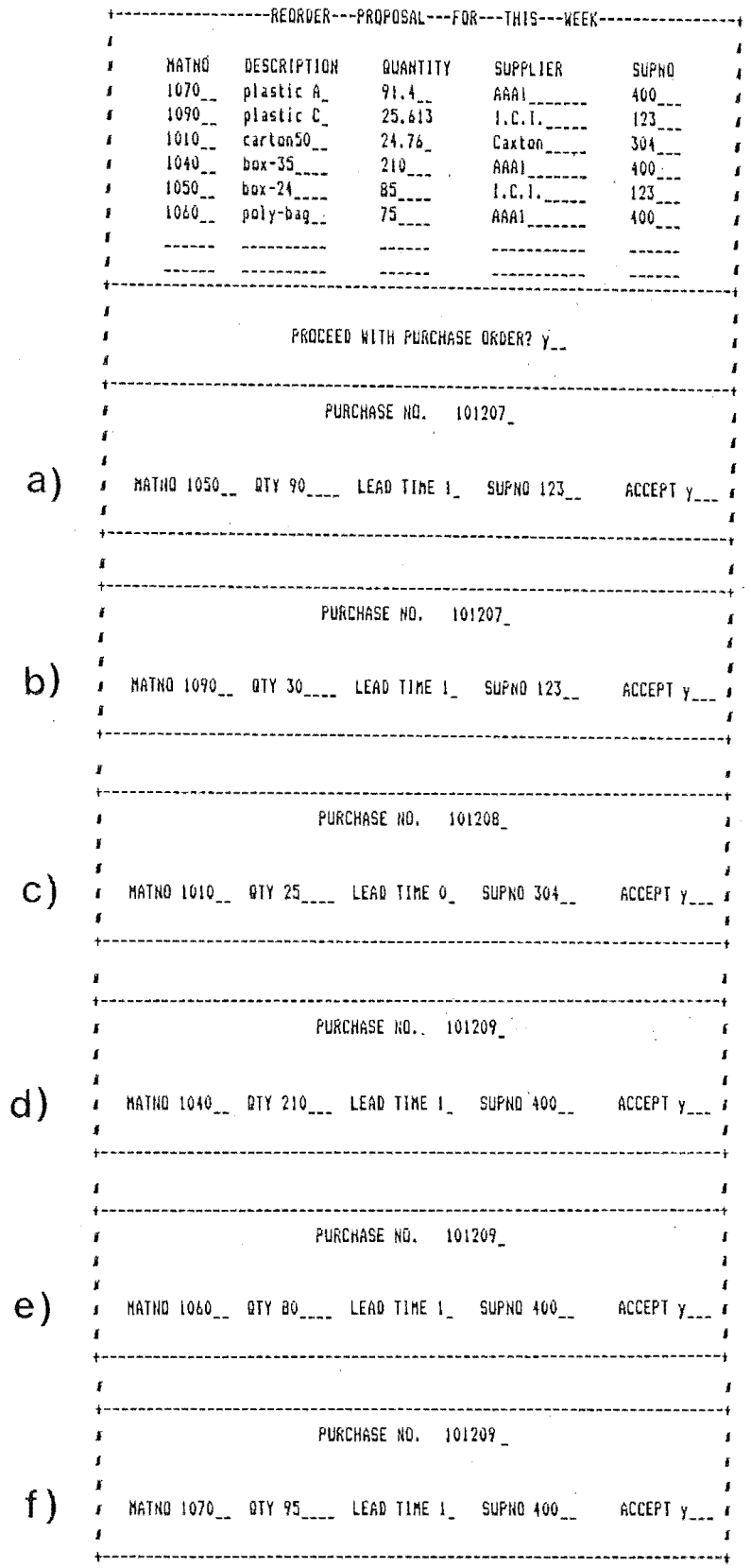


Figure J10 Material reorder proposals

```
SQL> select * from pur_sup;

DATESENT    PURNO   SUPNO DATEREC
--------- ------- ------- ---------
          101206
          101207     123
          101208     304
          101209     400
```

```
SQL> select * from purchase;

   PURNO   MATNO     QTY RECEIVED
------- ------- -------- --------
101207    1050       90
101207    1090       30
101208    1010       25
101209    1040      210
101209    1060       80
101209    1070       95
```

The ordered quantity values are inserted into the expected delivery or 'del' rows of the mrp table (boxed figures below).

```
+------------------------MATERIAL-REQUIREMENT-----------------------+
#                                                                   #
#  MATNO   DESCRIBE  WEEK0   WEEK1   WEEK2   WEEK3   WEEK4        #
#  1070__  req_____  ______  69____  87.4__  78.2__  64.4__       #
#  1070__  dev_____  ______  -5____  0_____  0_____  0_____       #
#  1070__  del_____  ______  70____  [95]___  0_____  0_____       #
#  1070__  or______  0_____  0_____  78.2__  64.4__  0_____       #
#  ______  --------  ------  ------  ------  ------  ------       #
#  1080__  req_____  ______  18.975  32.775  22.425  24.15_       #
#  1080__  dev_____  ______  .1____  0_____  0_____  0_____       #
#  1080__  del_____  ______  35____  35____  0_____  0_____       #
#  1080__  or______  0_____  0_____  28.225  0_____  0_____       #
#  ______  --------  ------  ------  ------  ------  ------       #
#  1090__  req_____  ______  31.913  43.7__  36.8__  32.2__       #
#  1090__  dev_____  ______  5_____  0_____  0_____  0_____       #
#  1090__  del_____  ______  45____  [30]___  0_____  0_____       #
#  1090__  or______  0_____  0_____  36.8__  32.2__  0_____       #
+-------------------------------------------------------------------+
#  EXIT:                                                            #
#                                                                   #
#  RESTORE COMMITED CAPACITY DATA   ___                             #
+-------------------------------------------------------------------+
```

Printing order forms

When page 2 (figure 35) is used to print out the order forms to be sent to the suppliers, the pur_sup table is updated. The date when the order forms are printed is inserted into the datesent column:

```
SQL> select * from pur_sup;

DATESENT    PURNO   SUPNO DATEREC
--------- ------- ------- ---------
28-MAY-90 101207     123
28-MAY-90 101208     304
28-MAY-90 101209     400
```

CONTROL form

Data entry for goods received

When goods are received, page 1 is used to enter the purchase number and the quantities received (figure 36). The purchase, pur_sup and mrp tables are updated [K7] (boxed figures below):

```
SQL> select * from purchase;

  PURNO   MATNO     QTY RECEIVED
------- ------- ------- --------
101207    1050      90       90
101207    1090      30       30
101208    1010      25
101209    1040     210
101209    1060      80
101209    1070      95
```

```
SQL> select * from pur_sup;

DATESENT    PURNO   SUPNO DATEREC
--------- ------- ------- ---------
28-MAY-90 101207     123 28-MAY-90
28-MAY-90 101208     304
28-MAY-90 101209     400
```

```
+------------------------MATERIAL-REQUIREMENT-----------------------+
#                                                                   #
#  MATNO    DESCRIBE   WEEK0    WEEK1    WEEK2    WEEK3    WEEK4   #
#  1070__   req_____  ______   69____   87.4__   78.2__   64.4__   #
#  1070__   dev_____  ______   -5____   0_____   0_____   0_____   #
#  1070__   del_____  ______   70____   95____   0_____   0_____   #
#  1070__   or______  0_____   0_____   78.2__   64.4__   0_____   #
#  ______   ________  ______   ______   ______   ______   ______   #
#  1080__   req_____  ______   18.975   32.775   22.425   24.15_   #
#  1080__   dev_____  ______   .1____   0_____   0_____   0_____   #
#  1080__   del_____  ______   35____   35____   0_____   0_____   #
#  1080__   or______  0_____   0_____   28.225   0_____   0_____   #
#  ______   ________  ______   ______   ______   ______   ______   #
#  1090__   req_____  ______   31.913   43.7__   36.8__   32.2__   #
#  1090__   dev_____  ______   [35]____   0_____   0_____   0_____   #
#  1090__   del_____  ______   [15]____   30____   0_____   0_____   #
#  1090__   or______  0_____   0_____   36.8__   32.2__   0_____   #
+-------------------------------------------------------------------+
#  EXIT:                                                            #
#                                                                   #
#  RESTORE COMMITED CAPACITY DATA   ___                             #
+-------------------------------------------------------------------+
```

Data entry for delivered orders

The orders that have been delivered to customers are recorded by using page 2 of the CONTROL form (figure 37). The out column (truncated to 'O'), in the orders table, is updated by the data entry. A 'y' character is inserted into the row of the order that has been delivered (boxed figure below).

```
SQL> select * from orders;

ORDERNO CUSTOMER   REQDATE   CO L PACTIME PSTART    PFINISH    PACKED O
------- ---------- --------- -- - ------- --------- --------- ------- -
    107 neworld    29-MAY-90 y  1    .215 28-MAY-90 28-MAY-90         [y]
   2002 france     29-MAY-90 y  o      .2 28-MAY-90 28-MAY-90
    108 w-mart     06-JUN-90 y  1    .044 04-JUN-90 04-JUN-90
   2003 england    06-JUN-90 y  o    .083 04-JUN-90 04-JUN-90
    109 farmer's   07-JUN-90 y  1    .917 28-MAY-90 04-JUN-90    .498
   2004 u.s.a.     13-JUN-90 y  o   1.188 11-JUN-90 12-JUN-90
```

PERFORMANCE form

Page 1 is used to calculate the material wastage index. The user inputs the weights of plastic materials that have been used in the USE field. The PRODUCE field shows the total weights of parts that are made from those materials. The calculation of (USE-PRODUCE)/USE gives the wastage value. The material wastage index is computed as the average of the WASTAGE values.

The other performance indices, as displayed in page 2 of the form (figure 39), are automatically computed when the MPS is rescheduled.

The moulding performance index is the ratio of the total 'actual' moulding time over total planned moulding time:

$$\frac{\text{sum (quantity moulded * moulding cycle time)}}{\text{sum (planned quantity * moulding cycle time)}}$$

The packaging performance index is the ratio of the total 'actual' packaging time over total planned packaging time:

$$\frac{\text{sum (quantity packaged * packaging time)}}{\text{sum (planned quantity * packaging time)}}$$

The calculation of the machines' utilisation indices:

$$\text{sum}\left(\frac{\text{quantity moulded in individual machine} * \text{moulding cycle time}}{\text{maximum capacity allowable in machine}}\right)$$

APPENDIX K

SQL*Forms Triggers

K1

```
[This trigger computes the
(maximum allowable capacity - planned capacity)
values for all the production processes.  It
also computes the MRP.]


select week4+7 into
:dummy1.y from week
insert into dummy1 (z)
values (1)
select z from dummy1 where z=5
update bom
set calc=0
update capacity
set (week4)=(select sum
(pactime*(op4+lp4)/3600) from mps)
where process=91
update bom
set (calc)=(select op4+lp4 from mps
where unitno=invno)
where invno between 9500 and 10000
update part
set (calc) = (select sum (bom.calc) from bom
where have = partno
and have between 500 and 700
group by have)
update bom
set (calc) = (select part.calc*cycle/3600 from
part
where partno=have)
where invno between 1 and 20
update capacity
set (week4)= (select sum(calc) from bom
where process=invno and
invno between 1 and 20
group by invno)
```

```
where process<90
update capacity
set week4=capa-week4
where capa is not null
update bom
set (calc)=(select part.calc*1.15 from part
where partno=invno)
where invno between 500 and 700
and have between 1070 and 1090
select :dummy1.y-7
into :dummy1.y
from week
select 'x' from orders
where pstart between :dummy1.y and :dummy1.y+4
update product
set (calc)=(select sum(pqty) from ord_prod,
orders
where pstart between :dummy1.y and :dummy1.y+4
and
ord_prod.orderno = orders.orderno and
ord_prod.prodno = product.prodno
group by ord_prod.prodno)
where prodno in (select prodno from
ord_prod,orders
where pstart between :dummy1.y and :dummy1.y+4
and ord_prod.orderno = orders.orderno)
update bom
set (calc)= (select product.calc from product
where invno=prodno)
where invno between 19500 and 20000
and have between 1010 and 1030
update bom
set calc=0
where invno between 19500 and 20000
and have between 1010 and 1030
update bom
set (calc)=(select (op4+lp4-oc4-lc4) from mps
where unitno=invno and op4+lp4>=oc4+lc4)
where invno between 9500 and 10000
and have between 1010 and 1030
update mrp
set (week4)=(select sum (bom.calc*qty) from bom
where have between 1010 and 2000
and have =matno group by have)
where describe='req'
update capacity
set week1=week4,week2=week1,week3=2,week4=week3
update mrp
set
week1=week4,week2=week1,week3=week2,week4=week3
where describe='req'
update mps set
op1=op4,oc1=oc4,lp1=lp4,lc1=lc4,
op2=op1,oc2=oc1,lp2=lp1,lc2=lc1,
op3=op2,oc3=oc2,lp3=lp2,lc3=lc2,
op4=op3,oc4=oc3,lp4=lp3,lc4=lc3
update dummy1 set z=z+1
```

```
update bom set (calc)=(select dev*bom.qty from
part
where prodno = invno and prodno between 500 and
700)
where invno between 500 and 700
update mrp set (week1)=(select week1-sum(calc)
from
bom where matno = have and have between 1070
and 1090
group by have ) where describe ='req' and matno
between 1070 and 1090
update mrp
set week1=0-week1,week2=0-week2,week3=0-
week3,week4=0-week4
where describe = any('dev,'del')
select min(matno) into :matno.x from mrp
update mrp
set (week1,week2,week3,week4)=(select
sum(week1),sum(week2),sum(week3),sum(week4)
from mrp
where describe=any ('req','dev','del')
and matno=:matno.x group by matno)
where describe='or' and matno = :matno.x
select :matno.x+10
into :matno.x from mrp
update mrp
set week1=0-week1,week2=0-week2,week3=0-
week3,week4=0-week4
where describe = any('dev,'del')
update mrp
set (calc1)=(select lot from material
where material.matno=mrp.matno)
where describe='or'
update mrp set
week2=week1+week2,week1=0
where describe='or'
and week1<calc1
update mrp set
week3=week2+week3,week2=0
where describe='or'
and week2<calc1
update mrp set
week4=week3+week4,week3=0
where describe='or'
and week3<calc1
update mrp set
week4=0
where describe='or'
and week4<calc1
update mrp set (calc1)=(select lead from
material
where mrp.matno = material.matno)
where describe='or'
update mrp set
week0=week1,week1=week2,week2=week3,
week3=week4,week4=0
where describe='or' and calc1=1
update mrp set
```

```
week0=week1+week2,week1=0,week2=0
where describe='or' and calc1=2
update mrp set
week1=week3,week2=week4,week3=0,week4=0
where describe='or' and calc1=2
delete from dummy1
where x is not null
update bom set (calc)=(select dev*cycle/3600
from part
where prodno = have and part between 500 and
700 )
where have between 500 and 700 and invno
between 1 and 20
update capacity set (week1)=(select
week1+sum(calc) from bom
where process = invno and invno between 1 and
20
group by invno ) where process<90
delete from dummy1 where z=5
delete from dummy1 where x is not null
```

K2

[This trigger processes the data of new orders and inserts the required data in the fields in page 2 of Creorder form.]

```
delete from ord_prod
where prodno is null
update product set calc=0
update product set (calc)=(select pqty from
ord_prod
where product.prodno=ord_prod.prodno and
orderno=:neworder.orderno)
where prodno in (select prodno from ord_prod
where orderno=:neworder.orderno)
update bom set calc=0
update bom set (calc)=(select pqty*qty from
ord_prod
where invno = prodno and
orderno=:neworder.orderno)
where invno in (select prodno from ord_prod
where orderno=:neworder.orderno)
update mps set calc1=0
update mps set (calc1)=(select sum(bom.calc)
from bom
where have = unitno and invno between 19500 and
20000
and have between 9500 and 10000 group by have )
update orders set (pactime)=(select
sum(calc1*mps.pactime)/
3600 from mps)
where orderno=:neworder.orderno
update bom set (calc)=(select mps.calc1*qty
from mps
where unitno = invno ) where
(invno between 9500 and 10000 and have between
500 and 700) or
(invno between 9500 and 10000 and have between
1040 and 1060)
update part set calc=0
update part set (calc)=(select sum(bom.calc)
from bom
where have = prodno and invno between 9500 and
10000 and
have between 500 and 700) where prodno between
500 and 700
update bom set (calc)=(select
part.calc*qty*1.15 from part
where prodno = invno and prodno between 500 and
700)
where invno between 500 and 700
update bom set (calc)=(select part.calc*cycle
from part
where prodno = have and prodno between 500 and
700)
```

```
where invno between 1 and 20
update capacity set (calc1)=(select
sum(bom.calc/3600) from bom
where invno = process and invno between 1 and
20
group by invno) where process <90
select sum(calc*vol) into :check.vol
from product
select sum(calc*wgt) into :check.wgt
from product
select sum(cost*bom.calc) into
:check.material_cost
from material , bom where have = matno and have
between 1010 and 2000 and matno between 1010
and 2000
select sum(calc1*overhead) into
:check.overhead_cost
from mps where unitno between 9500 and 10000
select sum(calc1*rates/3600) into
:check.moulding_cost
from capacity where process > 0
select pactime*rates into
:check.packlabour_cost
from orders, capacity
where orderno=:neworder.orderno and process=91
select :check.overhead_cost +
:check.material_cost +
:check.moulding_cost + :check.packlabour_cost
into :check.total_cost from dummy
select trunc((:check.vol/vol),0)+1 into
:check.container_and_no from container
where type='A'
select trunc((:check.vol/vol),0)+1 into
:check.container_b_no from container
where type='B'
select :check.container_and_no*vol-
:check.container_and_avail from container
where type='A'
select :check.container_b_no*vol-
:check.container_b_avail from container
where type='B'
```

K3

```
[This trigger loads orders into the
schedule and computes the material
availability and the uncommitted capacity
left in the schedule.]


delete from dummy1 where x='y'
update orders
set (pstart)=(select trunc((1-
sum(calc1)/sum(nvl(capa,0)))*5)+
:week.calc1 from capacity where process>89)
where orderno=:orderno.orderno
select 'x' from dummy where :load.w is null
update capacity set calc2=null
select 'x' from dummy where :orderno.l_o='1'
select 'x' from mps where lp_1>(lc_1+calc1)
update capacity set (calc1)=(select sum((lp_1-
lc_1-mps.calc1)*pactime)/3600 from mps where
lp_1>(lc_1+mps.calc1) and unitno
between  9500 and 10000)
where process=90
update capacity
set calc1=0 where process=90
select 'x' from mps
where lp_1<(lc_1+calc1)
update capacity
set (calc1)=(select capacity.calc1-
sum((lc_1+mps.calc1-lp_1)*pactime)/3600
from mps where lp_1<(lc_1+mps.calc1) and unitno
between 9500 and 10000)
where process =91
update capacity
set calc1=calc1-req
where process=91
update capacity
set calc1=calc1-req where process<90
update_machine_capa
update orders
set (pfinish)=(select trunc((1-
sum(nvl(calc1,0))/sum(nvl(capa,0)))*5)+
:week.calc1 from capacity where process >89)
where orderno=:orderno.orderno
update mps
set
op_2=null,oc_2=null,lp_2=null,lc_2=null,calc2=0
update week
set calc2=null
#exemacro exetrg twoweeks;

update mps set calc2=0
select 'x' from dummy where :orderno.l_o='1'
update mps set (calc2)=(select
trunc(capacity.calc1/req*mps.calc1,0) from
capacity where process =90)
where unitno between 9500 and 10000
```

```
update mps set calc2=calc1-calc2, calc1=calc2
update capacity set calc1=0
where process =90
update capacity
set (calc1)=(select capacity.calc1
-sum(mps.calc1 + lc_1)*pactime/3600 from mps)
where process=91
#exemacro exetrg lo_wk2;

select 'x' from mps
where lp_2>lc_2+calc2
update capacity set (calc2)=(select sum((lp_2-
lc_2-mps.calc2)*pactime)/
3600 from mps where lp_2>lc_2+mps.calc2 and
unitno between 9500 and 10000)
where process=90
update capacity
set calc2=0 where process=90
select 'x' from mps
where lp_2<lc_2+calc2
update capacity
set (calc2)=(select capacity.calc2-
sum((lc_2+mps.calc2-lp_2)*pactime)/3600 from
mps where lp_2<lc_2+mps.calc2 and unitno
between 9500 and 10000) where process =91
select 'x' from dummy

update mps set (calc2)=(select
trunc(capacity.calc1/req*mps.calc1,0) from
capacity where process =90)
where unitno between 9500 and 10000
update mps
set calc2=calc1-calc2,calc1=calc2
update capacity
set (calc1)=(select capacity.calc1-
sum(mps.calc1*pactime)/3600 from mps
where unitno between 9500 and 10000)
where process =91
update capacity
set (calc2)=(select capacity.calc2-
sum(mps.calc2*pactime)/3600 from mps
where unitno between 9500 and 10000)
where process =91
update bom set calc=0
update bom set (calc)=(select mps.calc1 from
mps
where unitno = invno)
where invno between 9500 and 10000
#exemacro exetrg update_part;

update part
set (calc)=(select sum(bom.calc) from bom
where have =prodno and have between 500 and 700
group by have )
update bom set (calc)=(select part.calc*cycle/
3600 from part where prodno = have )
where invno between 1 and 20
```

```
update capacity set (calc1)=(select calc1-
sum(calc) from bom
where process = invno and
invno between 1 and 20 group by invno)
where process<90
update bom set calc=0
update bom
set (calc)=(select mps.calc2 from mps
where unitno = invno )
where invno between 9500 and 10000
#exemacro exetrg update_part;
update capacity set (calc2)=(select calc2-
sum(calc) from bom
where process = invno and
invno between 1 and 20 group by invno)
where process<90

update orders
set (pfinish)=(select trunc((1-
sum(nvl(calc2,0))/sum(nvl(capa,0)))*
5)+ :week.calc2 from capacity where process
>89)
where orderno=:orderno.orderno
#exemacro exetrg load;

update mrp set calc2=0,calc1=0
update bom set calc=0
select 'x' from dummy
where :load.w is null
select 'x' from mps
where op_1+lp_1>oc_1+lc_1+calc1
update bom set (calc)=(select calc1+oc_1+lc_1
-(op_1+lp_1) from mps
where op_1+lp_1<calc1+oc_1+lc_1 and
invno=unitno)
where invno in (select unitno from mps where
op_1+lp_1<calc1+oc_1+lc_1)
#exemacro exetrg material;

update part set (calc)=(select
sum(bom.calc*qty) from bom
where prodno= have and invno between 9500 and
10000 and
have between 500 and 700 group by have ) where
prodno
between 500 and 700
update bom set (calc)=(select part.calc from
part
where prodno = invno and prodno between 500 and
700)
where invno between 500 and 700
update mrp
set (calc1)=(select sum(bom.calc*qty) from bom
where have = matno and have between 1040 and
1100
group by have )
where describe='req' and matno between 1040 and
1100
```

```
select 'x' from mps
where op_1+lp_1<calc1+oc_1+lc_1
update bom set (calc)=(select calc1+oc_1+lc_1-
(op_1+lp_1) from mps
where op_1+lp_1<calc1+oc_1+lc_1 and invno
=unitno)
where invno in (select unitno from mps where
op_1+lp_1<calc1+oc_1+lc_1
select 'x' from dummy where :orderno.l_o='1'
select 'x' from mps
where calc1=0 and lp_1<lc_1
update bom set (calc)=(select calc-(lp_1-lc_1)
from mps
where calc1=0 and lp_1>lc_1
and unitno = invno) where invno in (select
unitno from mps
where calc1=0 and lp_1>lc_1)
#exemacro exetrg material;
select 'x' from mps
where op_2+lp_2<calc2+oc_2+lc_2
update bom
set (calc)=(select calc2+oc_2+lc_2-(op_2+lp_2)
from mps
where op_2+lp_2<calc2+oc_2+lc_2 and
invno=unitno)
where invno in (select unitno from mps where
op_2+lp_2<calc2+oc_2+lc_2)
#exemacro exetrg material1;

update part set (calc)=(select
sum(bom.calc*qty) from bom
where prodno= have and invno between 9500 and
10000 and
have between 500 and 700 group by have ) where
prodno
between 500 and 700
update bom set (calc)=(select part.calc from
part
where prodno = invno and prodno between 500 and
700)
where invno between 500 and 700
update mrp
set (calc2)=(select sum(bom.calc*qty) from bom
where have = matno and have between 1040 and
1100
group by have )
where describe='req' and matno between 1040 and
1100

#exemacro exetrg cartons;

update bom set (calc)=(select pqty*qty from
ord_prod where orderno=:orderno.orderno and
prodno=invno) where invno in (select prodno
from ord_prod where
orderno=:orderno.orderno) and have between 1010
and 1030
```

```
update bom set (calc)=(select (op_1+lp_1-
(oc_1+lc_1))*qty from mps
where unitno=invno and unitno between 9500 and
10000)
where invno between 9500 and 10000
and have between 1010 and 1030
update material
set (calc)=(select sum(bom.calc) from bom
where have =matno and invno between 19500 and
20000
and have between 1010 and 1030 group by have )
where matno between 1010 and 1030
update material
set (calc)=(select material.calc-sum(bom.calc)
from bom
where matno=have and invno between 9500 and
10000
and have between 1010 and 1030 group by have )
where matno between 1010 and 1030
select 'x' from material where matno between
1010 and 1030 and calc>0
update mrp set (calc1)=(select material.calc
from material where material.matno=mrp.matno
and material.matno between 1010 and 1030
and material.calc>0) where matno in (select
material.matno from material where
matno between 1010 and 1030 and
material.calc>0)
and describe='req'
select 'x' from dummy
```

K4

```
[This trigger updates the stock levels
when data from the moulding processes
is entered.]


update part set dev=calc+dev,
actual=calc+actual
where calc is not null'
update bom set (calc)=(select nvl(part.calc,0)
from part
where prodno = invno and prodno between 500 and
700 )
where invno between 500 and 700
update material set (calc)=(select
sum(bom.calc*qty) from bom
where invno between 500 and 700 and matno =
have group by have ) where matno between 1070
and 1090
update mrp set (week1)=(select week1-
material.calc from material
where material.matno = mrp.matno and matno
between 1070 and 1090)
where describe = 'dev' and matno between 1070
and 1090
```

K5

```
[This trigger updates the stock levels and
the other tables which are affected by
the data entry of the packaging process.]


delete from dummy1 where x='y'
update ord_prod set
pack=nvl(pack,0)+nvl(calc,0)
where orderno=:packaging.orderno
update bom set calc=0
update bom set (calc)=(select
nvl(ord_prod.calc,0)*qty from ord_prod
where orderno=:packaging.orderno and
prodno = invno ) where invno in (select prodno
from ord_prod where orderno=:packaging.orderno)
update mps set calc=0
```

```
update mps set (calc1)=(select sum(bom.calc)
from bom
where have = unitno and invno between 19500 and
20000
and have between 9500 and 10000 group by have )
select 'x' from orders
where orderno=:packaging.orderno and l_o='l'
update mps set actual_l=actual_l+calc1
where unitno between 9500 and 10000
update mps set actual_o=actual_o+calc1
where unitno between 9500 and 10000
update bom set (calc)=(select mps.calc1*qty
from mps
where unitno = invno ) where invno between 9500
and 10000
update part set calc=0 where prodno between 500
and 700
update part set (calc)=(select sum(bom.calc)
from bom
where have = prodno and invno between 9500 and
10000 and
have between 500 and 700 group by have ) where
prodno between 500 and 700
update part set dev=dev-calc
where prodno between 500 and 700
update orders set (packed)=(select
sum(calc1*mps.pactime)/
3600/orders.pactime + nvl(packed,0) from mps)
where orderno=:packaging.orderno
update material set calc=0
where matno between 1010 and 2000
update material set (calc)=(select
sum(bom.calc) from bom
where have = matno and invno between 19500 and
20000
and have between 1010 and 1030 group by have )
where matno between 1010 and 1030
update material set (calc)=(select
sum(bom.calc) from bom
where have = matno and invno between 9500 and
10000
and have between 1040 and 1060 group by have )
where matno between 1040 and 1060
update mrp set (week1)=(select week1-calc from
material where
material.matno=mrp.matno and material.matno
between 1010 and 2000)
where describe = 'dev' and matno between 1010
and 1020
```

K6

```
[This trigger updates the various tables
when orders are released for the
purchase of materials.]


delete from dummy1 where x='y'
#exemacro case :update.lead is
when '0' then gostep lead0;
when '1' then gostep lead1;
when '2' then gostep lead2;
when others then null; end case;
update mrp set (week1)=(select :update.calc
from material
where material.matno=:update.matno and lead=0)
where matno = :update.matno and describe =
'del'
update mrp set (week2)=(select :update.calc
from material
where material.matno=:update.matno and lead=1)
where matno = :update.matno and describe =
'del'
update mrp set (week3)=(select :update.calc
from material
where material.matno=:update.matno and lead=2)
where matno = :update.matno and describe =
'del'
update mrp set week1=0 where
describe ='or' and matno = :update.matno
insert into purchase (purno,matno,qty)
select
:purchase.purno,:update.matno,:update.calc
from dummy
select 'x' from pur_sup
where purno=:purchase.purno
insert into pur_sup (datesent,purno,supno)
select :sysdate.sysdate,
:purchase.purno,:update.sup
from dummy
select 'x' from dummy
```